Reports of the Department of Geodetic Science
Report No. 190

THE OHIO STATE UNIVERSITY GEOMETRIC AND ORBITAL (ADJUSTMENT) PROGRAM (OSUGOP) FOR SATELLITE OBSERVATIONS

by

J. P. Reilly, C. R. Schwarz and M. C. Whiting

Prepared for

National Aeronautics and Space Administration
Washington, D.C.

Contract No. NGR 36-008-093
OSURF Project No. 2514

The Ohio State University
Research Foundation
Columbus, Ohio 43212

December, 1972

Reports of the Department of Geodetic Science

Report No. 190

THE OHIO STATE UNIVERSITY GEOMETRIC AND ORBITAL (ADJUSTMENT) PROGRAM (OSUGOP) FOR SATELLITE OBSERVATIONS

by

J. P. Reilly, C. R. Schwarz and M. C. Whiting

Prepared for

National Aeronautics and Space Administration
Washington, D. C.

Contract No. NGR 36-008-093
OSURF Project No. 2514

The Ohio State University
Research Foundation
Columbus, Ohio 43212

December, 1972

PREFACE

This project is under the supervision of Ivan I. Mueller, Professor of the Department of Geodetic Science at The Ohio State University, and is under the technical direction of James P. Murphy, Special Programs, Code ES, NASA Headquarters, Washington, D. C. The contract is administered by the Office of University Affairs, NASA, Washington, D. C., 20546.

TABLE OF CONTENTS

THE OSUGOP PROGRAM

1. INTRODUCTION

OSUGOP is a computer program that was developed at the Ohio State University. The name OSUGOP is an acronymn for Ohio State University Geometric and Orbital Program. The basic purpose of the program is to perform adjustments for ground station coordinates from observations made to satellites by stations observing from the ground. The observations can be optical or ranges, and the adjustments can be performed in either the geometric or the orbital mode. This program is based on many smaller programs developed in 1966 and 1967 and written in the SCATRAN language for use on the IBM 7094 computer [Krakiwsky, et al., 1967 and 1968]. Later programs were also written for use on the IBM 7094, but they were written in the FORTRAN IV language.

In the spring of 1969 the geometric adjustment of optical observations and the solutions of the normal equations programs were converted to the FORTRAN IV language from the original SCATRAN listings. At that time it was anticipated that further additions to the program would be necessary, and for this reason the programming was done in such a way that additional programs could be added very easily. A system of problem codes was established that would direct the computer to perform the different adjustments. In the fall of 1969 additional subroutines were added to the system to process range observations in the geometric mode.

From the fall of 1969 to the spring of 1971, the only changes made in OSUGOP were improvements in logic and additional constraint options. Then in the spring of 1971 subroutines were added to perform an orbital adjustment. An additional change was the ability to read optical data in the GEOS format.

Of the authors listed, C. R. Schwarz was a graduate student from September, 1967 to September, 1970, and is currently with the Defense Mapping Agency Topographic Center, Washington, D. C.; M. C. Whiting was a graduate student from September, 1970 to January, 1972, and currently lives in San Francisco, California.

2. PURPOSE OF THE PROGRAM

OSUGOP is an adjustment program that can be used for many different tasks. The main purpose is to perform an adjustment for observing station coordinates. The program, however, has been developed in such a way that certain specific tasks can be performed without resorting to a complete solution.

In order to control the data flow in the program, a system of "Problem Code Definitions" has been established. These codes are numbers punched in columns 1 through 20 of a data card that is read by the program near the beginning of the data deck. After the problem codes have been read the program uses these codes to branch to the subroutines needed to perform the required task. These "Problem Code Definitions" are given in Table 1. As it can be seen there are seven (7) different types of data that can be processed (see PCODE (1)). However, at the present time (September, 1972) only five of these are operational. The documentation included in Table 1 makes the table self-explanatory. If it is desired to perform a solution (i.e., PCODE (2) = 1), it is necessary to impose some constraints on the stations in the network. Table 2 is a listing of the "Constraint Code Directory." A complete description of each of these constraints is given in Section 4.

The purpose of this report is to describe <u>how</u> to use the OSUGOP program. In this case it is best to start with the arrangement of the card deck for each of the five possible types of adjustments designated by PCODE (1). Figures 1 through 5 are schematics of the various cases. Notice that in all cases the deck setup is the same through the station coordinate packet. Also notice that there are no program cards. The program itself is stored on a disk pack and the JCL cards at the beginning of the deck are all that is necessary to call the program.

Table 1

```
C               PROBLEM  CODE  DEFINITIONS
C
C      COLUMN   MEANING
C         1. OVERALL PROBLEM CODE
C              PCODE(1)=1  MEANS  OPTICAL  PROGRAM
C                       2  MEANS  RANGE
C                       3  MEANS  SOLUTION ONLY RUN
C                       4  MEANS ORBITAL MODE, OPTICAL OBSERVATIONS
C                       5  MEANS ORBITAL MODE, RANGE OBSERVATIONS
C                       6  MEANS ORBITAL MODE, MIXED OBSERVATIONS
C              PCODE(1)=7  MEANS OPTICAL PROGRAM, GEOMETRIC MODE(OLOS FORMAT)
C         2. PERFORM SOLUTION?
C              PCODE(2)=1  MEANS YES
C                       0  MEANS NO
C              PCODE(1)=3 IMPLIES PCODE(2)=1
C         3. MAXIMUM NUMBER OF ITERATIONS?
C              PCODE(1) MUST EQUAL 1 OR 2,
C              PCODE(2) MUST EQUAL 1,
C              PCODE(5) MUST EQUAL 1,  FOR ONE OR MORE COMPLETE ITERATIONS
C         5. FORM NORMALS?
C
C                PROCESSING CODES
C                                             1 MEANS YES,  0 MEANS NO
C         6. SIMULATE GUIDE MATRIX?
C         7. PRINT NORMALS?
C         8. PERFORM  SUMMARY BY OBSERVED LINES?
C         9. PUNCH NORMALS IN ASD FORMAT?
C        11. PRINT SATELLITE POSITION FOR EACH EVENT?
C              0 MEANS NO
C              1 MEANS PRINT XYZ AND GEODETIC COORDINATES
C              2 MEANS PRINT XYZ ONLY
C              3 MEANS PRINT GEODETIC COORDINATES ONLY
C        12. THIS PARAMETER DESCRIBES WHERE THE STANDARD DEVIATIONS OF THE
C              INDIVIDUAL OBSERVATIONS (USED TO FORM THE WEIGHTS) ARE TO BE FOUND
C              PCODE(12)=0 MEANS TO READ THE OBSERVATIONAL STANDARD DEVIATION
C              FROM THE CARD CONTAINING THE OBSERVATION.
C              PCODE(12)=1 MEANS TO ASSOCIATE A SINGLE STANDARD DEVIATION WITH
C              ALL OBSERVATIONS FROM A GIVEN STATION.** THE STANDARD DEVIATIONS
C              TO BE ASSOCIATED WITH EACH STATION ARE GIVEN IN COLUMNS 73-79 OF
C              THE CARD CONTAINING THE INPUT COORDINATES OF THE STATION.
C              PCODE(12)=2 MEANS TO ASSOCIATE A SINGLE STANDARD DEVIATION WITH
C              ALL OBSERVATIONS.** THIS NUMBER IS FOUND IN COLS. 21-30 OF THE
C              CARD CONTAINING THE TEST DISTANCE (OPTICAL) OR TEST VARIANCE
C              (RANGE).
C              ** IN THE CASE OF OPTICAL OBSERVATIONS, THIS NUMBER IS INTERPRETE
C              AS THE STANDARD DEVIATION OF THE DECLINATION AND OF THE RIGHT
C              ASCENSION TIMES THE COSINE OF THE DECLINATION, AND THE
C              COVARIANCE IS SET TO ZERO.
C
C         SOLUTION CODES
C
C        16. WRITE NORMALS AND INVERSE DURING SOLUTION PROCESSING?
C               0 MEANS PRINT NOTHING
C               1 MEANS PRINT PIVOT ELEMENTS
C               2 MEANS ALSO PRINT NORMALS AND INVERSE
C               3 MEANS ALSO PRINT REARRANGED NORMALS AND INVERSE

C        17. PUNCH ADJUSTED STATION XYZ AND VARIANCES FOR INPUT TO BADEKAS'
C               DATUM TRANSFORMATION PROGRAM?
C        18. PUNCH ADJUSTED STATION POSITIONS?
C        19. COMPUTE EIGENVECTORS OF VARIANCE-COVARIANCE MATRIX
C        20. COMPUTE CORRELATION COEFFICIENTS
C
```

Table 2

```
C
C          CONSTRAINT CODE DIRECTORY
C
C             WEIGHTED CONSTRAINTS
C
C  1  CONSTRAIN  THE COORDINATES OF A STATION AT A PRIORI VALUES*(I.E.WEIGHT IT
C  2  IMPOSE CHORD DISTANCE CONSTRAINT*.
C  3  IMPOSE RELATIVE POSITION CONSTRAINT*
C  4  IMPOSE DIRECTION CONSTRAINT*
C  5  CONSTRAIN THE GEODETIC LATITUDE,LONGITUDE AND HEIGHT OF A STATION.*
C
C            ABSOLUTE CONSTRAINTS
C
C 11  DEFINE THE ORIGIN OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C 12  DEFINE THE ORIENTATION OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT
C 13  DEFINE THE SCALE OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C 14  COMPLETELY FIX ONE OR MORE COORDINATES OF A STATION * $
C 15  COMPLETELY FIX ONE OR MORE COORDINATES  OF RELATIVE POSITION*$
C
C    *IF THE COORDINATES,RELATIVE POSITION, DISTANCE, OR DIRECTION, TO BE
C     CONSTRAINED ARE NOT GIVEN, THE CONSTRAINT IS COMPUTED FROM THE
C     APPROXIMATE COORDINATES OF THE STATION(S) INVOLVED
C
C     $THE DIAGONAL ELEMENTS OF THE W MATRIX ARE USED AS CODES TO INDICATE
C      WHICH COORDINATES ARE TO BE FIXED. A NON-ZERO CODE MEANS TO FIX
C      THE COORDINATE.
C
```

3. INPUT TO THE OSUGOP PROGRAM

The input is made up of card packets, which are groups of cards containing a variable number of cards, and signal cards. Each packet is terminated by an end signal card, which is blank in columns 1-79 and contains "E" in column 80. The one exception is the optical observation packet using the GEOS format. Here the "E" must be punched in column 73 (denoted "special end card" in Figure 2). Depending on the type of run, some packets may or may not be necessary.

3.1 Card Format for Required Cards

Title Packet (always required): As many title cards as desired are permitted, containing any text in columns 1-79. This text appears verbatum on the first page of the output. An end signal card terminates this packet.

Problem Codes (always required): These codes appear on a single card and control the type of processing to be performed by the program. See Table 1 for a description of each code and the column in which it is punched. (Do not put an end signal card after this card).

Datum Card Packet (always required): This contains a list of the ellipsoids on which the input and output ellipsoidal coordinates of the stations are located. Each datum is described by 2 cards.

Card 1.

Columns	Format	Contents
1-2	I2	Identifying number of datum.
3-15	F13.2	Semi-major axis of ellipsoid.
16-28	F13.2	Semi-minor axis of ellipsoid.

Card 2.

Columns	Format	Contents
1-32	4A8	32 character alphabetic name of datum.

The datum packet is terminated by an end signal card.

Station Coordinate Card Packet (always required): Each card gives the input (or approximate) coordinates of a station.

Columns	Format	Contents
1-4	I4	Identifying number of the station.
5-6	I2	Identifying number of the ellipsoid to which the ellipsoidal coordinates refer.
7-24	4A4, A2	18 character station name.
25	A1	Sign of latitude.
26-28	I3	Degrees of latitude.
29-31	I3	Minutes of latitude.
32-39	F8.4	Seconds of latitude.
40-42	I3	Degrees of longitude (+East).
43-45	I3	Minutes of longitude.
46-53	F8.4	Seconds of longitude.
54-63	F10.2	Ellipsoid height (in meters).
73-79	F7.2	Standard deviation to be used for all observations from this station (IF PCODE (12) = 1).

There is one card for each ground station in the network. This packet is terminated by an end signal card.

The identifying number of the datum for the station coordinates must correspond to the number defining the datum. For example, if one only has station coordinates on the North American Datum, the datum card packet could contain a card with the number 1 in column two, and the numbers 6378206.4 and 6356583.8 for the semi-major and semi-minor axes. Then on each station coordinate card one would have to put the number 1 in column six to show that these coordinates refer to the North American Datum.

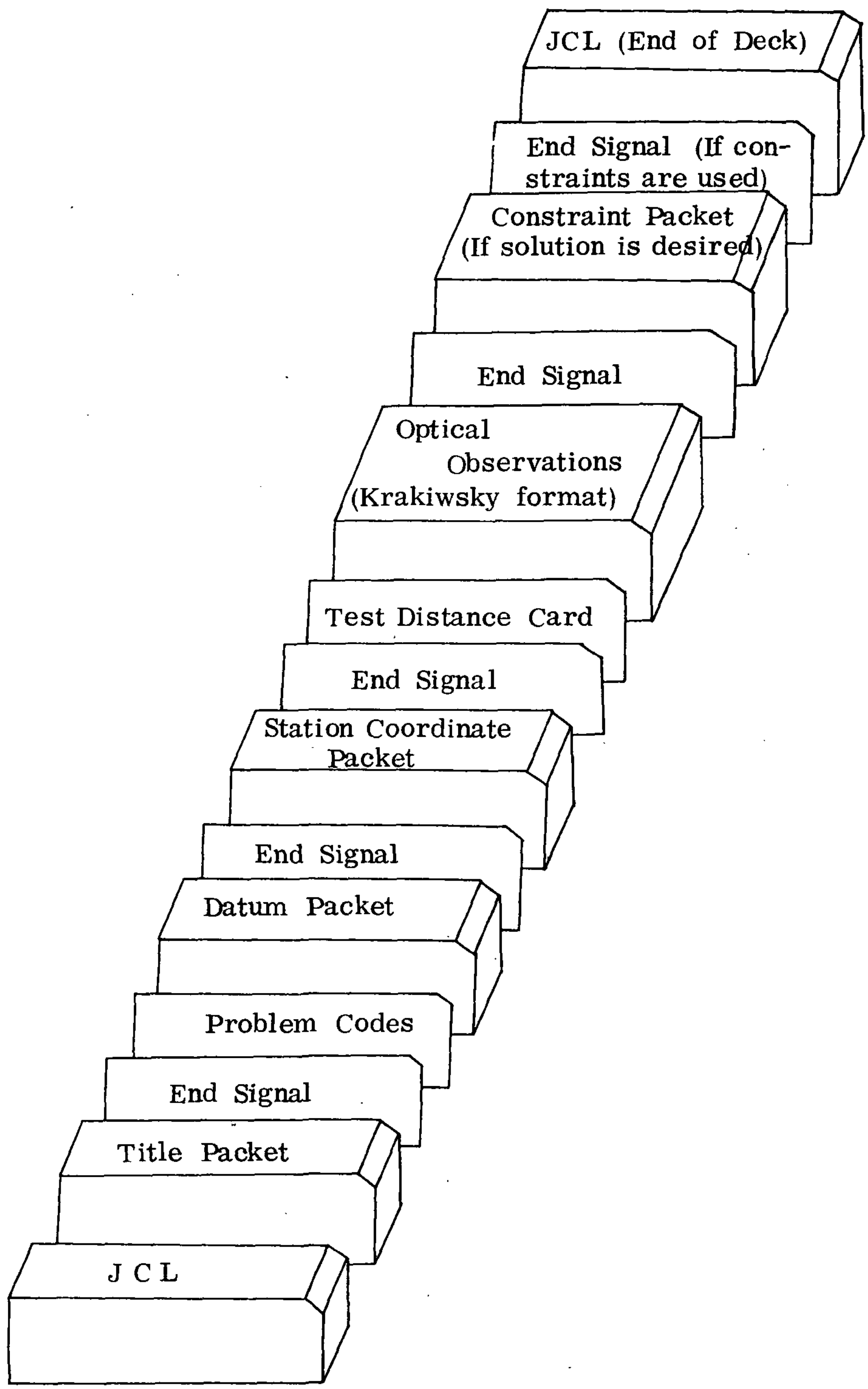


Figure 1. Deck Setup for Optical Program (OSU Format).

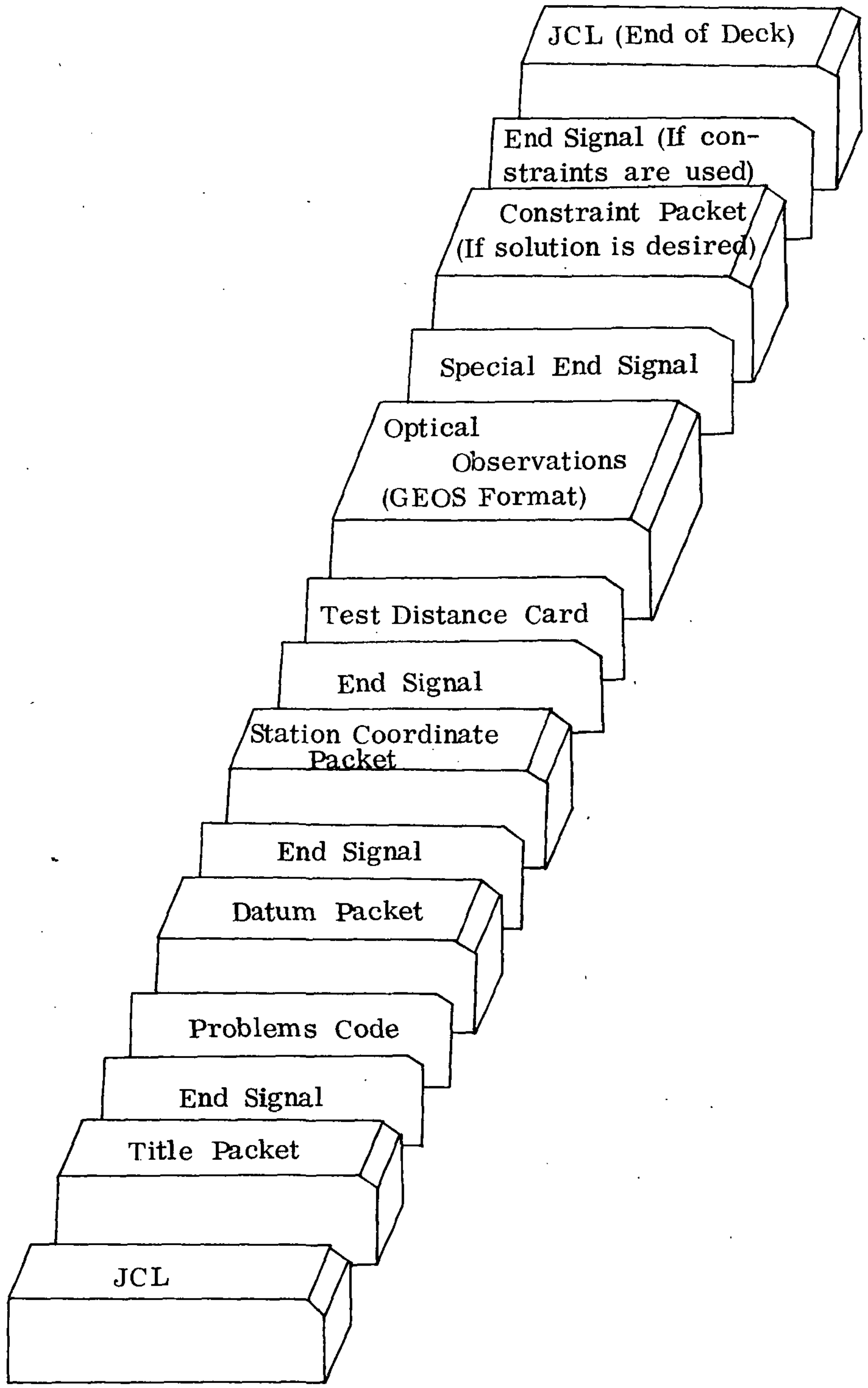


Figure 2. Deck Setup for Optical Program (GEOS Format).

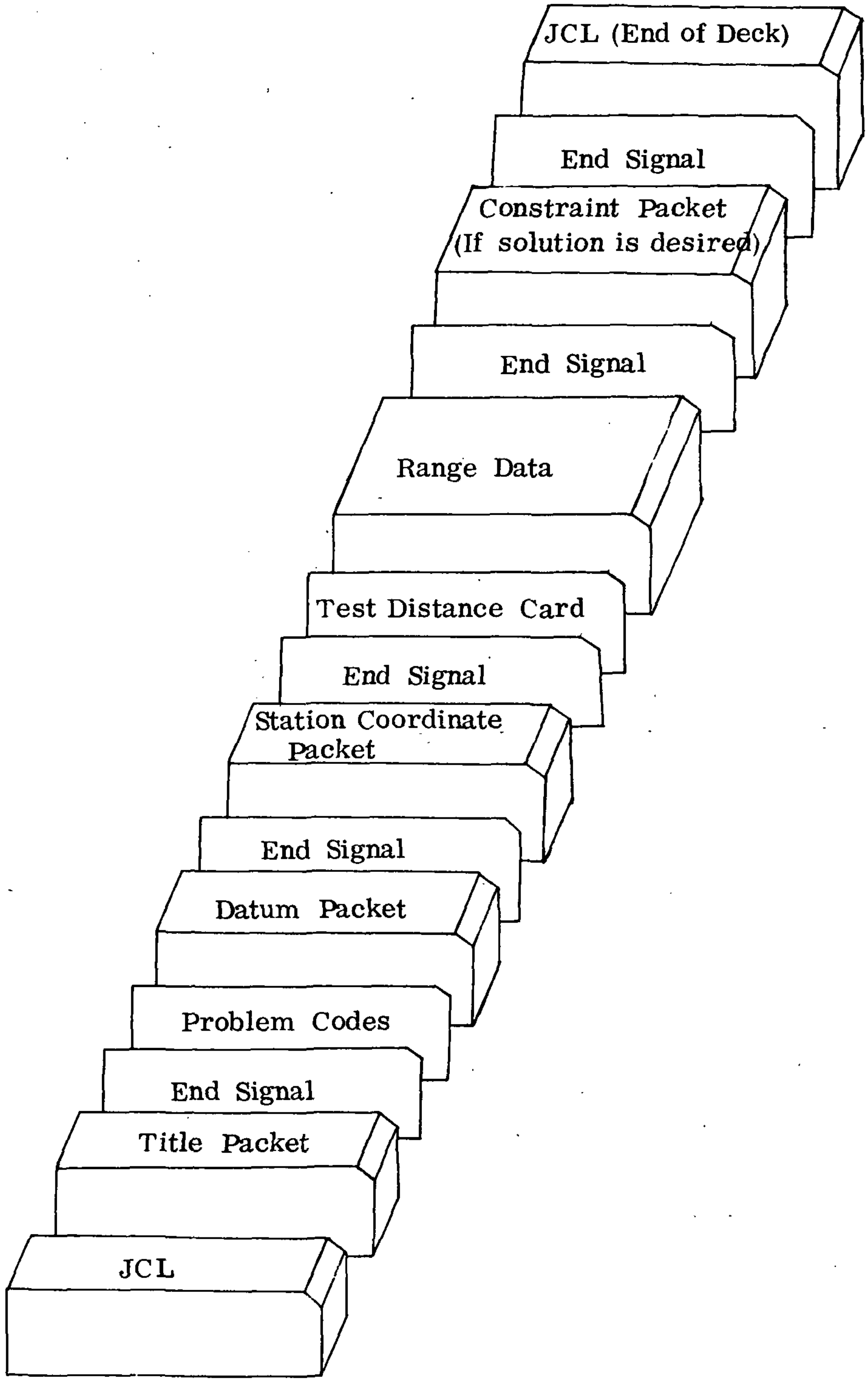


Figure 3. Deck Setup for Range Program in the Geometric Mode.

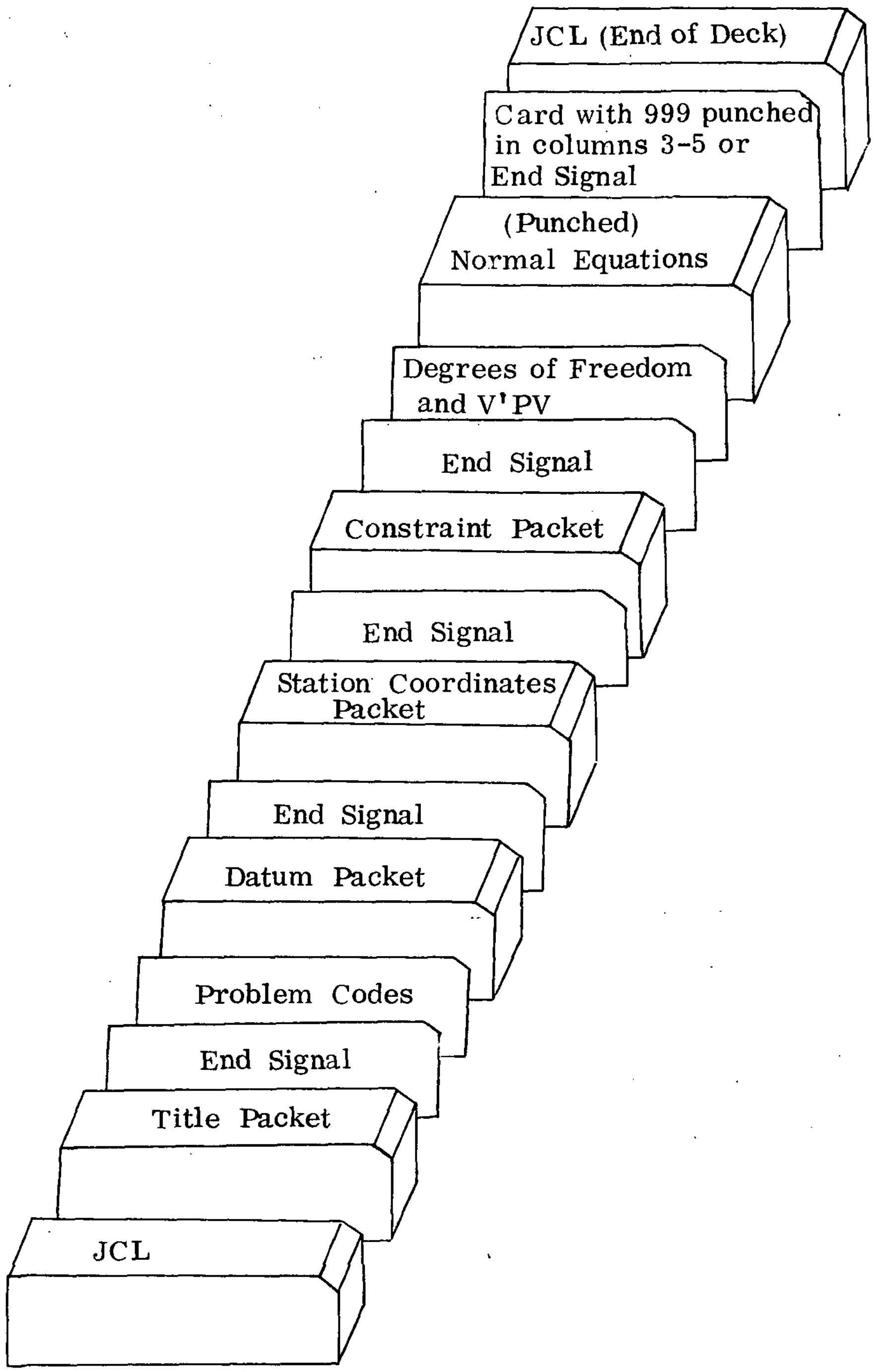


Figure 4. Deck Setup for Solution Using Normal Equations Punched on Cards.

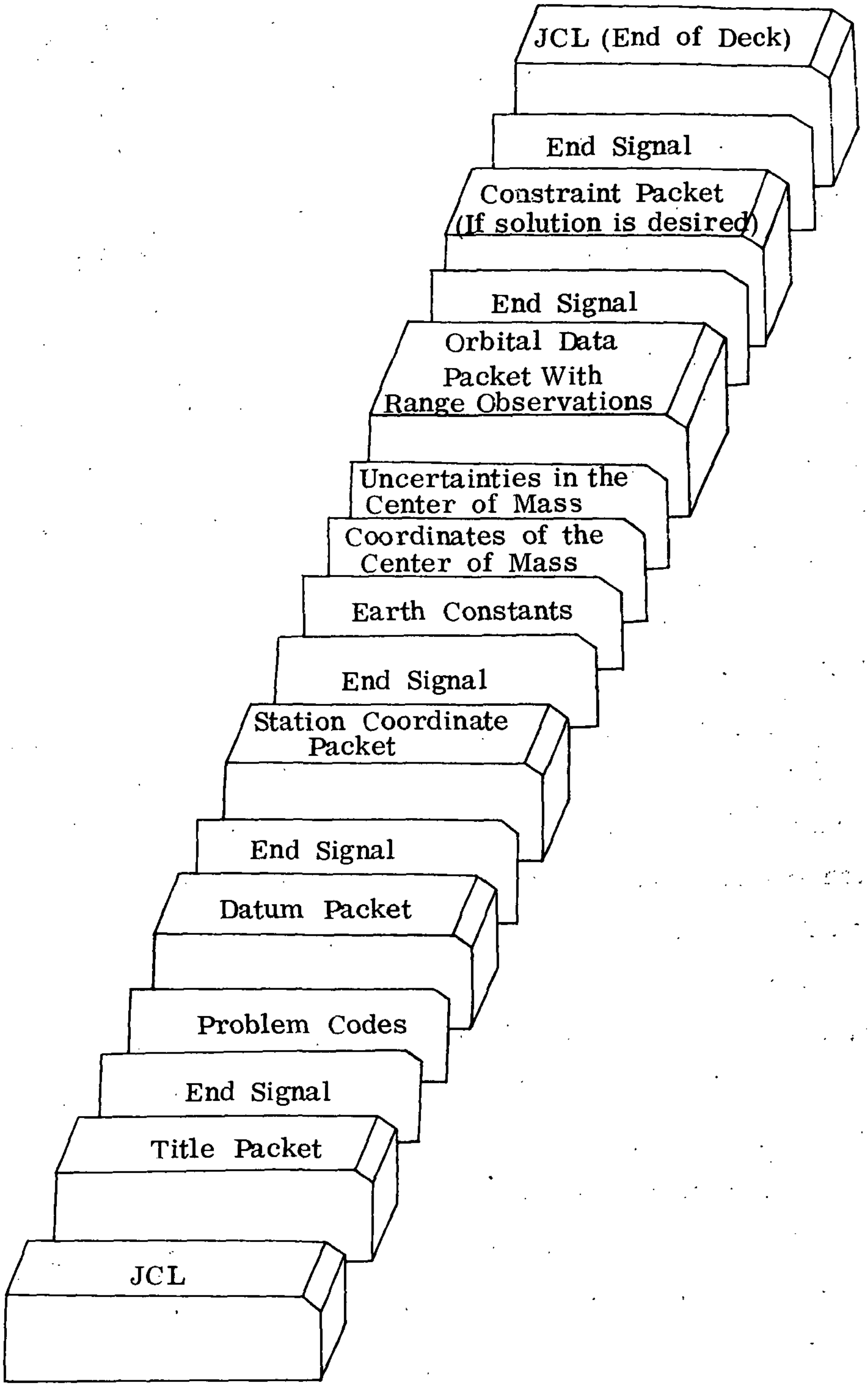


Figure 5. Deck Setup for Range Observations in the Orbital Mode.

3.2 Card Formats for the Different Options

There are five different types of input that can be handled by this program. The deck setup for each type of observation will be discussed separately.

3.2.1 Optical Observations, Geometric Mode.

There are two different card formats for the optical observations. However, the adjustment itself is the same for both. The reason for the two different formats is that the original optical programs written in SCATRAN utilized a card format that was convenient for the programmer and is referred to as the OSU format. Whenever the first optical data from the GEOS I satellite became available, it was easier to change the observations to the OSU format than to modify the program. When the optical program was later converted to FORTRAN IV the OSU format was retained. Because the GEOS format has been accepted as a standard format, the computer program was modified in 1971 to accept either card format.

As can be seen from Figures 1 and 2, it is necessary to have a test distance card in front of the optical observations. This test distance card is used to specify a rejection criteria for each observation. The purpose of the rejection criteria is to eliminate bad or questionable observations from the adjustment automatically without physically removing the observation cards from the card deck. The optical program is designed to read all the observations that have the same time of observation, and then perform an adjustment for the position of the satellite. The approximate station coordinates of the observation stations are held fixed during this adjustment. If the approximate coordinates of the observing stations are known to a certain degree of accuracy, and if the observations are known to a certain accuracy, the accuracy of the adjusted position of the satellite can be predicted.

The need to have a rejection criteria may not be very obvious, but from

past experience it has proved to be very useful. A fairly large percentage of the optical data received from the Space Science Data Center in Washington had large blunders. By setting the rejection criteria at a reasonable value, all bad observations were eliminated from the solution.

The rejection criteria can also be used when the data is reasonably free from bad data but some of the approximate station coordinates are not well known. Here the rejection criteria can be set fairly high for one or two iterations so that all observations are accepted and the questionable station coordinates are allowed to adjust.

In addition to specifying the rejection criteria, there is a place on the test distance card to insert a value for the standard deviations of all observations that will override the actual standard deviation punched on the observation cards. The standard deviations will then be the same for $\alpha \cos \delta$ and δ, and the covariance term will be zero. The need for this feature in the program became apparent when the standard deviations on the observation cards were noted to be completely out of line.

3.2.1.1 Arrangement of Optical Observations.

The only requirement in the arrangement of observation cards is that they are grouped by events. An event is all the data that has been observed on a satellite at the same instant of time. The time on the data cards should be the same, but the computer program will allow for a deviation of 0.0002 seconds. There must be at least two (2) observation cards in an event (i.e., a minumum of two stations must observe the satellite at the same time).

If there are more than two stations observing at the same instant of time the program will perform an adjustment for the satellite position starting with all observations. If any of these observations are bad the program will delete them and perform an adjustment with the remaining observations. If after deleting the bad observations there are less than two stations observing, the entire event is deleted.

The format of the optical observations must agree with the code punched on the problem code card. Using the OSU format PCODE (1) =1, and using the GEOS format PCODE (1) = 7. The card formats peculiar to these optical observations are described below:

Test Distance Card.

Columns	Format	Contents
1-20	F20.2	Rejection criteria, in seconds of arc, to be applied to each observation during editing.
21-30	F10.2	Standard deviation, in seconds of arc, to be used for all observations (if PCODE (12) = 2).

Optical Observations (OSU Format) Card Packet.

Columns	Format	Contents
1- 3	I3	Station identification number.
6-19	I2, I3, F9.4	Hours, minutes, seconds of observation (expressed in UT1).
20-26	I2, A3, I2	Day, month, year of observation (Note: month can be either three letters such as Jan, Feb, etc., or else the number 1,2,...,12).
27-41	2I3, F9.5	Hours, minutes, seconds of right ascension (α).
42-55	A1,I2, I3,F8.4	Sign, degrees, minutes, seconds of declination (δ).
58-62	F5.2	Standard deviation of α multiplied by the cosine of the declination, in seconds of arc.
63-67	F5.2	Standard deviation of δ, in seconds of arc.
68-72	F5.2	Covariance between $\alpha \cos \delta$ and δ, in seconds of arc squared.

This packet is terminated by an end signal card.

Optical Observations (GEOS Format) Card Packet

Columns	Format	Contents
15-18	I4	Station identification number.
19-24	3I2	Year, month, day of observation (Note: Month must be expressed as a number)
25-34	2I2, F6.4	Hours, minutes, seconds of observation (expressed in UT1).
35-44	I3, I2, F5.3	Hours, minutes, seconds of right ascension (α).
45-53	A1, 2I2, F4.2	Sign, degrees, minutes, seconds of declination.
72-74	F3.2	Standard deviation of α multiplied by the cosine of the declination, in seconds of arc.
75-77	F3.2	Standard deviation of declination δ, in seconds of arc.
78-80	F3.1	Covariance between $\alpha \cos \delta$ and δ, in seconds of arc squared.

This packet is terminated by a special end signal card. This card has the letter E punched in column 71.

3.2.2 Range Observations, Geometric Mode.

As with the optical observations, a test card is required in front of the range observations. The purpose of the card is the same as described for the optical observations. The arrangement of the range observations has the same basic requirement as optical observation, and this is the grouping of observations by events. The minimum number of observations for an event is four (4). This subprogram does not have the provision for elimination of individual observations from an event; if one observation is bad, the entire event is deleted.

When using the range observations in the geometric mode, PCODE (1) = 2. The card format for the range observations is the GEOS range format. The

GEOS format as given by NASA uses all 80 columns of the data card. Since many columns are used for information such as satellite number, year of launch, etc., they are not included in the card format description that follows. The only information included in this description is data pertinent to the observation itself.

Test Distance Card

Columns	Format	Contents
1-20	F20.2	Rejection criteria, in meters, to be applied to each event.
21-30	F10.2	Standard deviation, in meters, to be used for all observations (if PCODE (12) = 2).

Range Observation Card Packet

Columns	Format	Contents
15-18	I4	Station identification number.
19-24	3I2	Year, month, day of observation.
25-34	I2, I2, F6.4	Hours, minutes, seconds of observation.
44-53	F10.3	Range, in meters.
65-70	F6.3	Standard deviation, in meters.

This packet is terminated by an end signal card.

The time (hours, minutes, seconds) of observation for range observations in the geometric mode can be in any time system. In every other type of adjustment the times must be in the UT1 time system, but in this case, the times are used only to distinguish the different events.

3.2.3 Solution Using Punched Normal Equations.

Whenever the OSUGOP Program is used with observations, it is possible to punch the normal equations on cards. These normals are punched prior to the addition of constraints, which means that the matrix of the normal equations is singular. This punching is made possible by setting PCODE (9) = 1.

The reasons for punching the normal equations are many. The most common use is that if many different solutions are to be run using the same observations, but with different constraints it is more efficient to form the normals only once, and then run the different solutions by changing only the constraints. It may take fifteen minutes of computer time to form the normal equations, and only fifteen seconds to perform the solution. Another important reason for punching normal equations is for use in a combination solution of two or more different systems of equations. The different sets of normal equations can be solved together, using constraint or ties between the systems.

Because of the need for punched normal equations, the punched output should be in such a format that the user can easily distinguish which rows and columns refer to a particular station. This has been accomplished by forming the normal equations in what is referred to as the ASD format. This is a collapsed form of normal equations where all zero elements have been eliminated. All matrix elements are in the form of 3 x 3 matrices (3 unknowns for each station). However, additional information other than just the elements of the matrix is punched. It is necessary to include the stations numbers and a code number to indicate the end of a row. Also, the degrees of freedom (d.o.f.) and the summation of V'PV are needed. Since these normal equations are really a system of equations, the discrepency vector, U, must be included. The discrepancy vector is the vector U in the expressions

$$NX + U = 0$$

$$X = -N^{-1}U \, .$$

The punched output of normal equations starts by punching the d.o.f. and $\Sigma V'PV$ on one card, and then punching each row of the normal equation. The first card for each row is the station number. The next card gives the three elements of the U vector that corresponds to this station number.

The next three cards are the elements of the 3 x 3 matrix that correspond to this station. Following this are the off-diagonal elements, which in this case are the 3 x 3 matrices corresponding to all the other stations that co-observed with the first station of this row. These are denoted by punching a card with the station number, and then three cards with the 3 x 3 matrix. This is repeated for all stations that co-observed. After the last set of elements for the row have been punched, a card containing the number 999 is punched to indicate the end of the row. After that the next row is punched, and so on until the end of the matrix has been reached.

A sample of the punched output can be seen in Figure 6. The first line shows the d.o.f. (4069) and $\Sigma V'PV$ (6737.147170). The next printed line indicates that the first row of normal equations has station number 9066 on the diagonal. The following card gives the three elements of the U vector, and the three cards following this are the elements of the 3 x 3 matrix corresponding to station 9066. After the diagonal elements come the stations that co-observed with station 9066, and the 3 x 3 matrix of off-diagonal elements for each of these stations. In this case, they are stations 8015 and 8019. The end of the first row is marked by the 999 (15th line).

Whenever the computer is asked to punch a set of normal equations, it is always a good idea to set the proper PCODES to print the normals also, as well as a guide matrix to indicate the layout of the matrix. This can be done with PCODE (6) and PCODE (7) (see Table 1). The guide matrix that corresponds to the normal equations in Figure 6 is given below:

```
                         4069.000000      6737.147170
9066
  0.4463324330          -0.7695279730          -0.4691234316
  0.0441544859     0.0064557696    -0.0021695353
  0.0064557696     0.1576382832    -0.0109005340
 -0.0021695353    -0.0109005340     0.0317712453
8015
 -0.0077443357    -0.0022595720    -0.0029675092
 -0.0023964059    -0.0314680666     0.0096666831
 -0.0026492987     0.0092509108    -0.0125750867
8019
 -0.0164675851     0.0072840522    -0.0042015838
  0.0076429834    -0.0779199552    -0.0006815355
 -0.0018020912    -0.0010346222    -0.0058853776
999
8015
 -1.2612411155           0.1753680388           0.4569007003
  0.1472091420     0.0050456835    -0.0428749431
  0.0050456835     0.1041464281    -0.0037635258
 -0.0428749431    -0.0037635258     0.1294911894
8019
 -0.1198067022     0.0047556960     0.0421808712
  0.0045907000    -0.0467891914    -0.0123380813
  0.0427791438    -0.0057905394    -0.1058683524
9051
 -0.0106327398     0.0011620229     0.0030444112
  0.0011620229    -0.0001892168    -0.0005283645
  0.0030444112    -0.0005283645    -0.0014868776
999
9080
 -0.4712158212           0.0760527941           0.2145824528
  0.0162138738     0.0009955345    -0.0122346437
  0.0009955345     0.0046783024    -0.0012469197
 -0.0122346437    -0.0012469197     0.0242084646
999
```

Figure 6.

GUIDE MATRIX

9066	8015	8019	999
8015	8019	9051	999
9080	999		

Although it cannot be easily seen from the above guide matrix, the matrix of normal equations is upper-triangular. Another example of a guide matrix is shown in Figure 16.

When using punched normal equations to perform an adjustment, the deck setup is as shown in Figure 4. After the constraint packet, the card containing d.o.f. and $\Sigma V'PV$ is first, then the normal equations, and at the end the additional 999 card or an end-signal card with the letter E punched in Column 80. PCODE(1) is set equal to 3 in this case.

3.2.3.1 Combining Different Systems of Normal Equations.

If different systems of normal equations are to be combined for a single adjustment, the only additional work required is to physically combine the normal equations together into one data deck. When doing this, there are several things that must be done:

1. There can only be one card with degrees of freedom and $\Sigma V'PV$. Therefore, the values for each set of normal equations should be added together and the total d.o.f. and $\Sigma V'PV$ punched on one card.
2. There can only be one row for any one observing station. If station 9000 is included in more than one set of normal equations, the 3 x 3 matrices of diagonal elements corresponding to this station in each set of normal equations

must be added together to form one 3 x 3 diagonal matrix. The same goes for off-diagonal elements. If station 9000 co-observed with station 9010 in more than one set of normal equations, these 3 x 3 matrices must be added together to form one matrix.

3. Along the same line of reasoning mentioned in 2., if station 9000 is included in more than one set of normals, but in the later set co-observed with a station that was not included in the first set, the off-diagonal matrix corresponding to that station in the later set of normals must be moved into the row of the first set of normal equations. This can best be illustrated by guide matrices for two different systems:

First Set of Normal Equations

1	2	3	4	999
2	3	4	999	
3	4	999		
4	999			

Second Set of Normal Equations

4	5	6	999
5	6	999	
6	999		

Combined Set of Normal Equations

1	2	3	4	999
2	3	4	999	
3	4	999		
4	5	6	999	
5	6	999		
6	999			

4. Make sure that the matrix of normal equations is upper-triangular. As can be seen from the guide matrix of combined normals above, Station 1 co-observed with Station 2, but when one forms the second row, the 2-1 station combination is not repeated.

3.2.4 Range Observations, Orbital Mode.

In order to use range observations in the orbital mode, the deck set-up requires a great deal of work. It is necessary to have three data cards at the beginning of the data packet that give the earth constants, coordinates of the center of mass, and the uncertainties in the center of mass (see Figure 5). After this, the observations are separated into passes, and with each pass must be included the approximate orbital elements at a particular epoch and a code to tell the program what the coordinate system is. The epoch time is also included. Each pass is then separated by an end signal card. Because of the complexity of the deck setup, each step will be described in detail.

3.2.4.1 Earth Constants.

The earth constants are the semi-major axis of the earth, GM (or kM), gravitational constant × mass, and the rate of rotation of the earth. Also included on the earth constants card is the standard deviation of the observations if one wants to override the actual standard deviation punched on the observation cards. The card format is as follows:

Columns	Format	Contents
1-20	D20.8	Semi-major axis of the earth, in meters.
21-40	D20.8	GM, in units of cm^3/sec^2.
41-60	D20.8	Rotation rate of the earth, in radians/sec.
61-80	D20.8	Standard deviation, in meters, to be used for all observations (if PCODE (12) = 2).

3.2.4.2 Coordinates of the Center of Mass.

The coordinates of the center of mass give the location of the center of mass with respect to the origin of the ellipsoid used in the adjustment. The coordinates of the center of mass are given in the coordinate system in which the station coordinates are given.

	Columns	Format	Contents
Card 1	1-15	D15.8	X coordinate of the center of mass.
	16-30	D15.8	Y coordinates of the center of mass.
	31-45	D15.8	Z coordinate of the center of mass.
Card 2	1-15	D15.8	uncertainty of the X coordinate of the center of mass, in meters.
	16-30	D15.8	uncertainty of the Y coordinate of the center of mass, in meters.
	31-45	D15.8	uncertainty in the Z coordinate of the center of mass, in meters.

3.2.4.3 Orbital Data.

When inputing orbital data, the data is separated so that each orbit, or pass, is input separately. The first card of each pass gives the pass number, name and a code to indicate the type of preliminary orbital elements used to describe the orbit (see details below). The next two cards contain the orbital elements. The fourth card contains the time of epoch. After the time card, the observations for the particular pass are read. The cards do not have to be in any order, but to conserve computer time it is best to arrange the observations in the order of increasing time. The last card for the pass is an end signal card.

Each pass is arranged the same way as described above and placed one behind the other. There is no required order for arranging the different passes; the program works on each pass separately. After all passes are inserted in the deck, an extra end signal is placed after the last pass.

(Note: This end signal card is in addition to the one at the end of the last pass of data.)

The first card of the pass can have any number or name; these are used for identification purposes only. The code (IOCODE) used to indicate the type of preliminary orbital elements must be one of the numbers 0 through 4.

IOCODE = 0 - means rectangular elements are given in the True Sidereal System.

IOCODE = 1 - means rectangular elements are given in the Modified Sidereal System.

IOCODE = 2 - means the rectangular elements are given in the Earth-Fixed System.

IOCODE = 3 - means Keplerian elements are given, referred to the true equator.

IOCODE = 4 - means Keplerian elements are given, referred to the true equator and the 1950.0 equinox (i.e., the SAO Orbital System).

If the value of IOCODE is 0, 1 or 2, the orbital elements are expressed as X, Y, Z on one card, and $\dot{X}, \dot{Y}, \dot{Z}$ on a second card. If IOCODE is 3 or 4,, the orbital elements are expressed as the semi-major axis, eccentricity and inclination on the first card, and right ascension of the ascending node, argument of perigee, and mean anomaly on the second card.

The fourth card of each pass is the same regardless of the type of orbital elements used. This gives the epoch time, which is the time that corresponds to the orbital elements. This particular card, at first glance, appears to be very confusing due to the fact that there are several options for specifying epoch time. Figure 7 is a sample of the fourth card layout. If the value of Z CODE is left blank, it means that the orbital elements refer

EPOCH(MJD)	IDAY	MONTH	IYR	IH	MIN	ESEC	ZCODE	(IHR)	IMIN	SEC

Figure 7.

to the epoch time given at the left side of the card, and can either be expressed in MJD or day, month, year, hour, minutes, seconds. If the value of ZCODE is anything other than a blank, it means that the epoch time is the hours, minutes and seconds given on the right-hand side of the card. The distinction between these two times is that if the epoch time is given on the right-hand side of the card, the epoch time is outside the timespan of the pass, and the desired epoch time is the time given on the left side of the card. It means that the actual epoch time is that given to the right of ZCODE, which may be as much as 24 hours away from the time of the pass. In this case, the computer program updates the elements to the time given on the left side of the cards. Care must be taken to insure that the proper day is given, since only hours, minutes and seconds are given on the right-hand side of the card.

After the time card, the range observations for that particular pass are inserted with an end card placed after the last range card.

The card formats are as follows:

Orbital Data for Each Pass

	Columns	Format	Contents
Card 1	1-4	A4	Orbit number (can be anything, used for analysts identification only).
	5-52	6A8	Orbit name (for identification only, this can be left blank, if desired).
	53	I1	IOCODE. This is the number 0, 1, 2,3 or 4 depending on the coordinate system of the orbital elements.

(i) Orbital elements given in rectangular coordinates (IOCODE = 0, 1 or 2).

Card 2	1-15	D15.8	X coordinate of satellite, in meters.
	16-30	D15.8	Y coordinate of satellite, in meters.
	31-45	D15.8	Z coordinate of satellite, in meters.

	Columns	Format	Contents
Card 3	1-15	D15.8	$\dot{X}$, the velocity component in the X direction, in meters/sec.
	16-30	D15.8	$\dot{Y}$, the velocity component in the Y direction, in meters/sec.
	31-45	D15.8	$\dot{Z}$, the velocity component in the Z direction, in meters/sec.

(ii) Orbital elements given as Keplerian elements (IOCODE = 3 or 4).

	Columns	Format	Contents
Card 2	1-15	D15.8	Semi-major axis of orbital ellipse, in meters.
	16-30	D15.8	Eccenticity of orbital ellipse.
	31-45	D15.8	Inclination of ellipse to equatorial plane, in degrees and decimal degrees.
Card 3	1-15	D15.8	Right ascension of the ascending node, in degrees and decimal degrees.
	16-30	D15.8	Argument of perigee, in degrees and decimal degrees.
	31-45	D15.8	Mean anomaly, in degrees and decimal degrees.

Epoch Time Card

	Columns	Format	Contents
Card 4	1-15	D15.8	Epoch time expressed in modified Julian days (MJD).
	16-20	I5	Day of the month.
	23-25	A3	Month of the year. This can be the numbers 1 through 12 or the first three letters of the month's name, such as JAN, FEB., etc.
	26-30	I5	Year (last two digits).
	31-35	I5	Hours.
	36-40	I5	Minutes.
	41-50	D10.5	Seconds.

If the orbital elements given are at a time not in the actual pass itself, this time is punched as follows:

Columns	Format	Contents
51-55	I5	Hours
56-60	I5	Minutes
61-70	D10.5	Seconds

(Note: In order for the computer to use this time, something other than blanks must be in columns 54 or 55. Therefore, even if the time is zero hours, the zeros must be punched in columns 55 or 54 and 55).

It should be noted that on the format of card No. 4, there is no ZCODE as such described. The FORTRAN coding has been done in such a way that columns 54 and 55 are recognized as ZCODE and also as part of the hour value.

Observations Cards.

The format for the observation card for the orbital adjustment is identical to the observation card format described in the Range Observations, Geometric mode section of this report (3.2.2).

The end of data on each pass is marked by placing an end signal card after the last observation.

4. CONSTRAINTS

Table 2 is the directory of the constraint codes needed to apply constraints to normal equations prior to a solution. There are ten different types of constraints that can be applied, five types of weighted constraints and five types of absolute constraints. In all cases, the first card gives the constraint code, and then depending on the type of constraint, the cards following give the required information necessary to apply the constraint.

4.1 Weighted Constraints.

4.1.1 Constrain the Coordinate of a Station at its a Priori Value.

This constraint is used to weight any one or all three Cartesian coordinates of a station. It is used primarily to control the translation or to define the origin of a network of stations. The weight needed to apply this constraint is

$$W = \frac{\sigma_o^2}{\sigma_i^2}$$

where

σ_o^2 is the a priori unit variance (which, in most cases, is assumed to be 1).

σ_i^2 is the variance of the component of the station coordinate, in meters squared.

Four cards are needed to apply this constraint. The first card is the constraint code, which in this case is 1. The second card is the number of the station to be constrained. On the third card are listed the coordinates to be constrained, and the fourth card gives the weights to be applied to each of the coordinates.

If the coordinates to be constrained are the approximate coordinates given at the beginning of the program, the third card is replaced by a blank.

The card formats are as follows:

	Columns	Format	Contents
Card 1	1- 2	I2	Constraint code, which in this case is the number 1.
Card 2	1- 5	I5	Station number of the station to be constrained.
Card 3	1-16	D16.8	The X component of the station coordinate, in meters.
	17-32	D16.8	The Y component of the station coordinate, in meters.
	33-48	D16.8	The Z component of the station coordinate, in meters.
			Any one, or all three of the coordinates can be constrained. If only one or two components are to be constrained, let the field blank for the part not to be constrained. If the approximate coordinates as given at the beginning of the program are to be constrained, card 3 should be blank.
Card 4	1-16	D16.8	Weight to be applied to the X component.
	17-32	D16.8	Weight to be applied to the Y component.
	33-48	D16.8	Weight to be applied to the Z component.

4.1.2 Chord Distance Constraint.

The chord distance constraint is used primarily to apply a scale. The chord distance is computed by the program if the approximate station coordinates are to be used to compute the chord. If the chord distance is known from another source, the distance is punched onto a card.

Three cards are needed to apply this constraint. The first card is the constraint code, which is 2. The second card gives the station numbers of the two stations involved in the chord constraint. The third and last card gives the chord distance, and the accuracy of the distance. If the

accuracy of the chord distance is 1 part in 500,000, the number punched on the card is 500,000. The card formats are as follows:

	Columns	Format	Contents
Card 1	1- 2	I2	Constraint code, which is 2.
Card 2	1- 5	I5	Station number of the first station.
	6-10	I5	Station number of the second station.
Card 3	1-16	D16.8	Chord distance (in meters). This is left blank if the chord distance is to be computed from the approximate coordinates.
	17-32	D16.8	Accuracy of chord distance, expressed as the denominator of the accuracy ratio.

4.1.3 Relative Position Constraint.

This constraint is used when two or more stations have known positions with respect to each other. After the adjustment, the relative positions of these stations should remain unchanged, or the change should be within the limit of accuracy of the survey that tied the stations together. A common use for this constraint is where two or more stations are observing from the same small island where the positions of the stations are known on a local datum survey but the positions on the datum of the adjustment are not known.

The relative position constraint can only be applied between two stations at a time. If there are more than two stations involved, additional relative positions constraints must be used. As an example, if the relative positions between stations 1, 2 and 3 are to be constrained, a constraint can be applied between stations 1 and 2, and an additional constraint between stations 2 and 3. A third constraint can be applied between stations 1 and 3, but it isn't necessary.

Four cards are needed to apply this constraint. The first card is

the constraint code, which is 3. The second card gives the station numbers of the two stations involved in the relative position constraint. The third card gives the ΔX, ΔY, ΔZ coordinate difference between the two stations that is to be constrained during the adjustment. The fourth and last card gives the weights of the coordinate differences. A word of caution is necessary about the sign convention. The signs of the coordinate differences on card three must correspond to the order the station numbers appear on card two. As an example, if on card two the station numbers are 1 and 2, with 1 being punched in the first field, the sign convention for card three must be $X_1 - X_2, Y_1 - Y_2$, and $Z_1 - Z_2$.

The card formats are as follows:

	Columns	Format	Contents
Card 1	1-2	I2	Constraint code, which is 3.
Card 2	1-5	I5	Station number of first station (1).
	6-10	I5	Station number of second station (2).
Card 3	1-16	D16.8	Coordinate difference ΔX, expressed as $X_1 - X_2$.
	17-32	D16.8	Coordinate difference ΔY, expressed as $Y_1 - Y_2$.
	33-48	D16.8	Coordinate difference ΔZ, expressed as $Z_1 - Z_2$.
Card 4	1-16	D16.8	Weight to be applied to the ΔX coordinate difference.
	17-32	D16.8	Weight to be applied to the ΔY coordinate difference.
	33-48	D16.8	Weight to be applied to the ΔZ coordinate difference.

If the approximate coordinates are to be used to compute ΔX, ΔY and ΔZ, card 3 should be left blank.

4.1.4 Direction Constraint.

When the direction between two stations i and j is to be constrained, it can be accomplished by applying weights to two angles α and β defining the direction between them. These angles are defined as

$$\alpha = \tan^{-1} \frac{\Delta Y}{\Delta X}$$

$$\beta = \tan^{-1} \frac{\Delta Z}{R}$$

where

$$\Delta X = X_i - X_j$$

$$\Delta Y = Y_i - Y_j$$

$$\Delta Z = Z_i - Z_j$$

and

$$R = (\Delta X^2 + \Delta Y^2)^{\frac{1}{2}}$$

As with some of the other constraints, if the directions are to be computed from the approximate station coordinates, it is not necessary to precompute α and β.

Four cards are needed to apply this constraint. The card formats are as follows:

	Columns	Format	Contents
Card 1	1-2	I2	Constraint code, which is 4.
Card 2	1-5	I5	Station number of first station.
	6-10	I5	Station number of second station.
Card 3	1-16	D16.8	Alpha (α), in seconds.
	17-32	D16.8	Beta (β), in seconds.
Card 4	1-16	D16.8	Standard deviation of α, in seconds of arc.
	17-32	D16.8	Standard deviation of β, in seconds of arc.
	33-48	D16.8	Covariance term, in seconds of arc^2.

If the approximate coordinates are to be used to compute α and β, Card 3 should be left blank.

4.1.5 Constraint on Geodetic Latitude, Longitude and Height.

This constraint can be applied to the latitude, longitude and height or to any one of the three. The main use for this constraint has been to apply height constraints to island stations where the orthometric height has been well determined, and the separation between the geoid and the reference ellipsoid is known to a certain degree of accuracy. It can also be used to define the origin of a network. This is identical in concept to constraint code 1 except here the coordinates constrained are φ, λ, h.

Four cards are needed to apply this constraint. The card formats are as follows:

	Columns	Format	Contents
Card 1	1-2	I2	Constraint code, which is 5.
Card 2	1-5	I5	Station number of the station to be constrained.
Card 3	1-16	D16.8	Latitude φ, in degrees and decimal degrees.
	17-32	D16.8	Longitude λ, in degrees and decimal degrees.
	33-48	D16.8	Height h, in meters.
Card 4	1-16	D16.8	Standard deviation of φ, in seconds of arc.
	17-32	D16.8	Standard deviation of λ, in seconds of arc.
	33-48	D16.8	Standard deviation of h, in meters.

Any one, or all three of the coordinates can be constrained. If only one or two components are to be constrained, leave the field blank for the part not to be constrained. If the approximate coordinates are to be constrained, Card 3 should be blank.

4.2 Absolute Constraints

The five absolute constraints are listed in Table 2. Three of these constraints use the inner adjustment equations, and for this reason a more detailed description is necessary [Blaha, 1971].

Whenever an adjustment is to be performed on a network of observing stations, it is necessary to define an origin, establish some form of orientation, and set a scale. With optical observations, the orientation is determined from the observations themselves, and with range observations the scale is determined from the observations. The inner adjustment constraint package was developed for use when the origin, orientation or scale was not known. An example of its use could be on a net of observing stations, each station on an isolated island in the ocean. If the observing stations were cameras, the origin and scale would have to be determined before adjustment. By applying constraint codes 11 and 13, the program would use the inner adjustment equations to get the best origin and scale possible from the geometry of the network and the observations themselves.

Only one card is necessary to call any one of the inner adjustment constraints. This is the same as the first card of the weighted constraint package, which is the code number punched in columns 1 and 2 of the card. If the origin is to be defined, use code 11; for orientation, code 12; for scale, code 13. Codes 14 and 15 are not operational, but the same results can be obtained by using constraints 1 and 3, using very large weights.

4.3 Using Constraints Only in an Adjustment.

In addition to the five different types of adjustment (the deck set-ups

of which are described in Figures 1 thru 5) it is possible to perform an adjustment without observations. This can be done by using constraints only. If there are enough constraints applied to tie all the stations together, this is equivalent to forming a set of normal equations. In this computer program, the constraints are always added to the existing normal equations (see [Mueller, et al., 1970], pp. 10-16 for a description of this). If the existing normal equations do not exist, the normal equations can be formed entirely from constraints. Care must be taken to insure that all stations are constrained properly.

The deck set-up for solution using constraints only is shown in Figure 8. This is identical to the deck set-up for a solution only (see Figure 4) except the degrees of freedom card and the punched normal equations are replaced by a blank card. As with the solution only run shown in Figure 4, PCODE (1) must be set equal to 3 on the problem codes card.

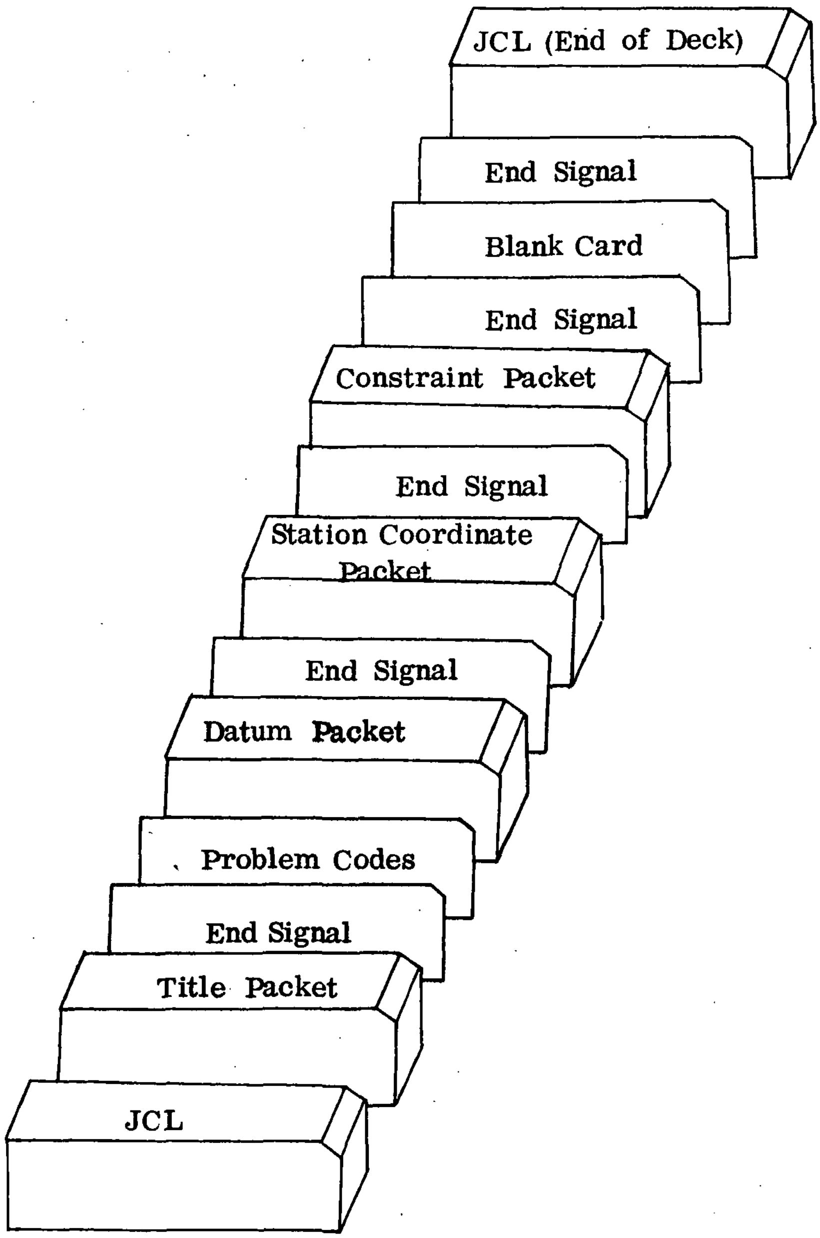


Figure 8. Deck Setup for Solution Using Constraints Only.

5. OUTPUT

As far as the geodetic analyst is concerned, the most important part of an adjustment is the final results which are given in the printed output from the computer. However, there is more information included in this printout than the solution vector itself, and for this reason an explanation may be necessary. In this section of the report, the entire printed output is explained in detail.

5.1 Output Common to All Adjustments.

The first few pages printed on any adjustment are identical in format regardless of the type of adjustment. These pages contain the information input to the program on punched cards giving the description of the job itself, the problem codes, the datums involved in the adjustment, and the input coordinates of stations. Figures 9, 10, and 11 are samples of the actual output for each of the items mentioned. It should be noted that at the bottom of Figure 11, there is written 'TEST DISTANCE = 5.00 SECONDS OF ARC'. This one line will differ for the different types of runs. The above is written for optical data. With range data in the geometric mode it will be 'TEST VARIANCE = some number', and for range observations in the orbital mode, it will be a printout of the coordinates of the center of mass and the uncertainties of these coordinates.

5.2 Output from Geometrical Adjustment.

5.2.1 Output of Optical Observations.

When running the optical adjustment, all events are printed out as shown in Figure 12. As can be seen, the iteration number is printed, then the test distance and then for each of the events, the adjustment information. This is referred to as an event adjustment because it is truly an adjustment for the satellite position. There will be a maximum of six lines of information for each 2 station event, as is illustrated in Figure 12.

SECOR SOUTHWEST PACIFIC DATA

THE GEOCENTRIC COORDINATES OF THE MAUI BAKER-NUNN IS CONSTRAINED AT 7 METERS IN EACH COMPONENT (X,Y,Z)

THE ELLIPSOIDAL HEIGHTS OF STATIONS 1 TO 7 ARE CONSTRAINED TO VALUES DETERMINED BY USING THE SAO GEOID MAP TOGETHER WITH THE LEVELED HEIGHTS AT EACH STATION

THE EGRS 6 DATA IN QUAD 3-8-10-11 FROM AMS IS INCLUDED

PROBLEM CODES 21201100000200010111

Figure 9.

DATUMS INVOLVED IN ADJUSTMENT

DATUM 1 SAC STANDARD EARTH 1969 A= 6378155.00 METERS B= 6356770.10 METERS

Figure 10.

```
                                       INPUT COORDINATES OF STATIONS

9001        ORGAN PASS, N.M.                  DATUM   1      SAO STANDARD EARTH 1969
          GEODETIC COORDINATES            32 25 24.7246       +253 26 48.3689    1611.5300
          CARTESIAN COORDINATES      -1535756.999      -5166995.995      3401042.003

9007        AREQUIPA, PERU                    DATUM   1      SAO STANDARD EARTH 1969
          GEODETIC COORDINATES         - 16 27 57.1295        +288 30 24.1819    2476.0400
          CARTESIAN COORDINATES       1942775.000      -5804081.001     -1796933.004

9009        CURACAO, ANTILLES                 DATUM   1      SAO STANDARD EARTH 1969
          GEODETIC COORDINATES            12  5 24.7586       +291  9 44.1872     -27.9300
          CARTESIAN COORDINATES       2251828.999      -5816918.998      1327160.003

               TEST DISTANCE =                  5.00    SECONDS OF ARC
```

Figure 11.

```
               BEGIN ITERATICN      1

             TEST DISTANCE =                 5.00    SECONDS OF ARC

EVENT      1
  9007   9 20 27.88550     9AUG65 17  3 15.9510   - 4 51 32.9900      2.00  2.00  0.0        0.4
  9009   9 20 27.88550     9AUG65 16 46 40.8540   -39 42 38.7800      2.00  2.00  0.0        0.3
 SATELLITE PCSITION      909591.839   -9777429.886   -2145935.965
 GEOD. COORD. OF SATELLITE    -12.378295    275.314914     3674216.8
         GQI=    0.32985    RMS MISCLOSURE IN METERS=       8.8

EVENT      2
  9007   9 29 59.89010     9AUG65 16 49 43.6620    36 45  5.2200      2.00  2.00  0.0        1.4
  9009   9 29 59.89010     9AUG65 16 34 17.2960     1 27 20.2100      2.00  2.00  0.0        1.7
 SATELLITE POSITION      606301.075   -9926959.236    1439667.748
 GEOD. COORD. OF SATELLITE      8.271373    273.495068     3671402.6
         GQI=    0.33681    RMS MISCLOSURE IN METERS=      37.5

EVENT      3
  9007   7 26  0.00370    12AUG65 21 21 29.6330    31 53 42.2100      2.00  2.00  0.0        0.1
  9009   7 26  0.00370    12AUG65 21 10 44.5540   - 6 37  8.1500      2.00  2.00  0.0        0.1
 SATELLITE POSITION     5212034.937   -8549156.629     859722.350
 GEOD. CCORD. OF SATELLITE      4.928418    301.368777     3671504.9
         GQI=    0.38921    RMS MISCLOSURE IN METERS=       1.6
```

Figure 12.

The first output line for each event is the number of the event. This numbering starts at 1 for the first event and continues on. The second line is the observational data from the first station in the event, plus the residual in seconds after the adjustment for the satellite position. Referring again to Figure 12, event 1, line 1, the information as one reads across the line is:

9007	Station Number.
9 20 27.8855 =	$9^h 20^m 27^s.8855$ UT1
9 Aug 65	Date
17 3 15.9510 =	$17^h 3^m 15^s.9510$ Right Ascension
-4 51 32.9900 =	$-4^\circ 51' 32''.9900$ Declination
2.00	Standard deviation in right ascension, multiplied by the cosine of the declination, in seconds of arc.
2.00	Standard deviation in declination, in seconds of arc.
0.00	Covariance between $\alpha\cos\delta$ and δ, in seconds of arc, squared.
0.4	The residual, in seconds of arc, after the adjustment for the satellite position.

There will be one line of information for each observation. The information printed on lines 4 and 5 of event 1 give the satellite position in XYZ coordinates plus the geodetic coordinates φ, λ, h of the satellite. Either, or both, or neither of these two lines can be printed if the analyst so desires. This output is controlled by the value used for PCODE (11) (see Table 1).

The last line of each event gives a term referred to as GQI, and the RMS misclosure in meters. The term GQI stands for Geometric Quality Index, and is just the determinant of the matrix of normal equations used

in the event adjustment divided by the number of stations in the adjustment. It is used to give an idea of the conditioning of the matrix of normal equations; the smaller the GQI, the better the conditioning.

The events listed in Figure 12 are excellent examples of good data. However, not all data is good, and several examples of this are shown in Figure 13. As was mentioned earlier in this report, the optical program can reject observations and still give a satisfactory adjustment provided that after all rejections there are still good observations from at least two stations. In Figure 13, events 2279 and 2280 each have one observation rejected, which is denoted by the * printed at the end of the printed line. In both cases, the other two observations were good and the events were acceptable. At the bottom of Figure 13, events 2310 and 2311 were deleted due to insufficient number of good observations, which is the meaning of KODE = 2. If KODE = 3, it means that the deletion was due to insufficient geometrical separation between observations.

5.2.2 Output of Range Observations, Geometric Mode.

The output for the range observations, geometric mode, is almost identical in format to that of the optical observational data described earlier. A sample output is shown in Figure 14. The minimum number of stations required is 4. The adjustment for the satellite position is a least squares adjustment that iterates until convergence (maximum of 20 iterations). Referring to the second line of event 2, Figure 14, the information as one reads across the line is:

5401	Station Number.
66 July 3	Date (Notice that the order of the year and the day are the reverse of the optical printout.)
1 31 43.9990	$1^h 31^m 43^s.9990$ time
2164169.973	Range, in meters.
3.20	Standard deviation of the range measurement.
-1.12	The residual, in meters, of the range observation after the adjustment for the satellite position.

```
EVENT  2279
 8015     0 11 23.96570     17MAR66 14 43 34.5510     23 48 44.6300         2.00  2.00   0.0        0.9
 9066     0 11 23.96570     17MAR66 14 20 59.5640     18 19 12.8700         2.00  2.00   0.0        0.9
 9004     0 11 23.96570     17MAR66 16 10 23.4030     29 55 54.4000         2.00  2.00   0.0       13.1*
SATELLITE POSITION     6328333.525     2143166.810     5475604.534
GEOD. COORD. OF SATELLITE      39.474837      18.709293      2268926.8
          GOI=    0.01753   RMS MISCLOSURE IN METERS=         11.3

EVENT  2280
 8015     2 11 59.96550     17MAR66 11  1 14.5560     23 47 31.5500         2.00  2.00   0.0        0.7
 9004     2 11 59.96550     17MAR66 12  0 35.4290     48 29 26.5100         2.00  2.00   0.0       18.6*
 9066     2 11 59.96550     17MAR66 11  7 57.7680     17 44 50.4200         2.00  2.00   0.0        0.6
SATELLITE POSITION     6503593.505    -1269492.559     5543847.906
GEOD. COORD. OF SATELLITE      40.056820     348.954827      2270261.5
          GOI=    0.01183   RMS MISCLOSURE IN METERS=          9.0


EVENT  2310
 8015     3 20 20.02100     17OCT66 10 24 42.5850     59 19 54.6300         2.00  2.00   0.0       11.3*
 9051     3 20 20.02100     17OCT66 23 36 24.5590     70  5 58.0500         2.00  2.00   0.0        9.2*
          GOI=    0.58580   ENTIRE EVENT DELETED, KODE=   2

EVENT 2311
 8015     3 20 24.02100     17OCT66 10 30  9.5860     59  7  7.2500         2.00  2.00   0.0       11.4*
 9051     3 20 24.02100     17OCT66 23 31 24.4550     70 43 59.3900         2.00  2.00   0.0        9.4*
          GOI=    0.58163   ENTIRE EVENT DELETED, KODE=   2
```

Figure 13.

```
                BEGIN ITERATION      1

             TEST VARIANCE =                  10.00


EVENT      2
   5401   66  JUL   3     1 31 43.9990        2164169.973           3.20          -1.12
   5402   66  JUL   3     1 31 43.9990        3213575.617           3.20          -2.06
   5403   66  JUL   3     1 31 43.9990        2235385.881           3.20           1.23
   5404   66  JUL   3     1 31 43.9990        2777136.727           3.20           1.81
      VARIANCE OF EVENT ADJUSTMENT =            1.00  AFTER  5 ITERATIONS

EVENT      3
   5401   66  JUL   3     1 34 20.0010        2420777.136           3.20           0.21
   5402   66  JUL   3     1 34 20.0010        2565181.074           3.20           0.27
   5403   66  JUL   3     1 34 20.0010        2158629.200           3.20          -0.19
   5404   66  JUL   3     1 34 20.0010        2304264.963           3.20          -0.25
      VARIANCE OF EVENT ADJUSTMENT =            0.02  AFTER  5 ITERATIONS
```

Figure 14.

For the two events given in Figure 14, notice that PCODE (11) must have been zero (0) because of the fact that no satellite positions are printed. The last line of each event gives the variance of the event adjustment σ_0^2 (σ_0 = V'PV/d.o.f.), in meters squared. Also, the number of iterations of the least squares adjustment is given.

5.2.3 Output Common to All Geometric Adjustments.

After the events are printed in either the range or optical geometric adjustments, there are quite a few options, and all of these options are controlled by the PCODES. The basic options of course are; do you want to form normal equations? If you do form normal equations, do you want to perform a solution? Then there are the secondary options; do you want to print the normals? Do you want to punch the normals? Do you want to simulate the guide matrix? Do you want to perform a summary by observed lines? There are also several solution codes that are controlled by PCODE (16) thru PCODE (20). The analyst would be well advised to reread Table 1 to see just how all these options are initiated.

If there are no normal equations formed and no solution to be performed, the words 'NORMAL TERMINATION' are printed and the program stops. If normal equations are formed, and if a solution is to be performed, the first set of information that is printed is the analysis of misclosures by station and a summary of information as shown in Figure 15. It should be noted that these are the values prior to the addition of the constraints.

If PCODE (6) = 1, the guide matrix is printed. This is just a matrix to show at a glance what stations co-observed, and the arrangement of the matrix of normal equations. Figure 16 is a sample guide matrix. The number 999 printed at the end of each line in Figure 16 is just an indication of the end of a row. When the normal equations are generated in the computer, the number 999 is used to indicate the end of a row.

ANALYSIS OF MISCLOSURES BY STATION

STATION	NUMBER OF OBSERVATIONS	RMS MISCLOSURE
9001	0	0.0
9007	16	0.31
9009	16	0.31

TOTAL NUMBER OF GOOD OBSERVATIONS	32
TOTAL NUMBER OF GOOD EVENTS	8
CORRESPONDING DEGREES OF FREEDOM	8
TOTAL SUM OF SQUARES OF MISCLOSURES	3.06
CORRESPONDING STANDARD DEVIATION OF UNIT WEIGHT	0.62
WPW INCLUDING CONTRIBUTION FROM SATELLITE POSITION (I.E., VPV+UX)	2.96

Figure 15.

GUIDE MATRIX

5401	5401	5402	5403	5404	999				
5402	5402	5403	5404	5405	5406	5407	999		
5403	5403	5404	5405	5406	5407	5408	5410	5411	999
5404	5404	5405	5406	5407	999				
5405	5405	5406	5407	999					
5406	5406	5407	5408	999					
5407	5407	5408	5410	5411	999				
5408	5408	5410	5411	999					
5410	5410	5411	999						
5411	5411	999							

Figure 16.

If PCODE (7) = 1, the normal equations are printed. The description of the normal equations was described in Section 3.2.3. The printout would be in the format as shown in Figure 6.

If the inner adjustment constraints are used, the computer will print out one or more of the following messages:

"THE ORIGIN OF THE COORDINATE SYSTEM IS DEFINED BY INNER ADJUSTMENT PROCEDURE."

"ORIENTATION OF THE COORDINATE SYSTEM DEFINED BY INNER ADJUSTMENT PROCEDURE."

"SCALE OF THE COORDINATE SYSTEM DEFINED BY INNER ADJUSTMENT PROCEDURE."

Another option available is the summary by observed lines, which can be had by setting PCODE (8) = 1. This is just a summation of the actual number of observations made by every two station combination in the network. Figure 17 is a sample listing.

If a solution is to be performed, a set of constraints must be included in the data deck as described in Section 4. When the constraints are included, they are printed out after the summary of observed lines. Figure 18 is a sample of a partial listing. It is just a printout of the

	OBSERVATIONS ON EACH LINE	
2	6	27
2	7	14
2	8	23
2	9	19
2	10	22
6	7	31
6	8	23
6	9	19
6	10	21
7	8	14
7	9	27
7	10	21
8	9	14
8	10	23
9	10	27

Figure 17.

```
THE ELLIPSOIDAL COORDINATES (LAT.,LONG.,HEIGHT) OF STATION      36
ARE CONSTRAINED AT

38.893352222 DEGREES       267.795033329 DEGREES          306.000 METERS

   ON DATUM    1           NORTH AMERICAN

THE WEIGHTS FOR THESE CONSTRAINTS ARE COMPUTED FROM OBSERVATIONAL STANDARD DEVIATIONS OF

   0.100 SECONDS           0.100 SECONDS                  5.000 METERS


CHORD DISTANCE CONSTRAINT IMPOSED BETWEEN STATION   27
                                      AND STATION   29
CONSTRAINED DISTANCE=     1531562.90
THE WEIGHT IS COMPUTED FROM A RELATIVE UNCERTAINTY OF  ONE PART IN        750000.00
```

Figure 18.

original constraints, but with enough printed titles and labels to make the printed output easily understood by anyone.

The description of the results after adjustment will be deferred until Section 5.4.

5.3 Output from Orbital Adjustment, Range Data.

The output from an orbital adjustment differs in every respect from the geometric adjustment, except for the final parameter output for each station. This is necessary because in essence each event of a geometric adjustment corresponds to one pass of orbital information, and naturally orbital passes contain many more observations, etc.

The first group of output common to all orbital adjustments is the input information itself as shown in Figure 19. This information is printed for every pass in the data deck.

The second group of output data is the results of the adjustment of each pass for the first iteration (see Figure 20). Notice that the orbital elements printed at the beginning of each pass are the Apparent Celestial Cartesian Coordinates. Regardless of the original set of orbital elements, the program converts to this system for the adjustment. If the original orbital elements are very close to the actual elements, the resulting misclosures will be very small. However, the usual case is that the approximate orbital elements will cause fairly large misclosures (Figure 20 is a typical example). However, if the observations are good, the second iteration will have very small residuals.

A good point to keep in mind when examining this particular portion of the printout is that the program has no way of rejecting bad observations in a pass, and even if an entire pass is bad, the program cannot reject it. It is up to the analyst to examine the misclosures for each observation as well as the RMS misclosures for the pass, and to physically remove from the data deck observations which are bad, or even entire

```
EARTH CONSTANTS FOR ORBIT INTEGRATION
   SEMI-MAJOR AXIS          GM                  ROTATION RATE
     6378155.000      3.98601300000D 14      0.72921151470D-04


2        PASS 2

X=    -0.58395216D 07 Y=    -0.32876645D 07 Z=     0.40715531D 07
XDOT=  0.17910145D 04 YDOT= 0.35914915D 04 ZDOT= 0.58180899D 04
EPOCH= 0.40249399D 05

VALUES STORED ON UNIT 3

X=    -0.58395216D 07 Y=    -0.32876645D 07 Z=     0.40715531D 07
XDOT=  0.17910145D 04 YDOT= 0.35914915D 04 ZDOT= 0.58180899D 04
EPOCH= 0.40249399D 05                  28  JAN   69
                      GAST= 0.47290896D 01
```

STATION	DATE	TIME(UT)	OBSERVED RANGE	UNCERTAINTY
4082	28 JAN 69	9 34 21.7000	2505094.64	2.00
4082	28 JAN 69	9 34 41.6998	2488615.81	2.00
4061	28 JAN 69	9 34 16.2000	2381823.98	2.00
4061	28 JAN 69	9 34 36.1999	2480479.17	2.00
4061	28 JAN 69	9 34 56.2000	2581702.52	2.00
4860	28 JAN 69	9 35 19.3793	1918628.62	2.00
4860	28 JAN 69	9 35 39.3791	1872374.74	2.00
4860	28 JAN 69	9 35 59.3791	1834132.55	2.00
4082	28 JAN 69	9 35 1.8000	2478700.29	2.00
4082	28 JAN 69	9 35 21.7998	2475526.67	[illegible]
4082	28 JAN 69	9 35 41.7999	24790[illegible]	[illegible]
4061	28 JAN 69	9 35 16.1999	[illegible]	2.00
4061	28 JAN 69	9 35 36.2000	[illegible]56.26	2.00
4061	28 JAN 69	9 35 5[illegible]	[illegible]383015.10	2.00
4860	28 JAN 69	9 3[illegible]	3470256.36	2.00
4860	28 JAN 69	[illegible].1892	3017246.32	2.00
4860	28 JAN [illegible]	[illegible] 31.1888	3127873.84	2.00
[illegible]	[illegible]	[illegible]41 51.1987	3239424.59	2.00
[illegible]	[illegible]	9 42 19.3769	2646703.51	2.00
4860	28 JAN 69	9 42 39.3762	2742840.17	2.00
4860	28 JAN 69	9 42 59.3759	2841280.66	2.00
4082	28 JAN 69	9 42 1.8000	3555865.76	2.00
4082	28 JAN 69	9 42 21.8000	3651257.54	2.00
4760	28 JAN 69	9 42 11.1885	3351769.98	2.00
4760	28 JAN 69	9 42 31.1877	3464788.52	2.00
4760	28 JAN 69	9 42 51.1876	3578380.17	2.00

Figure 19.

BEGIN ITERATION 1

2 PASS 2 EPOCH= 28 JAN 69 9H 33M 56.0000S UT=MJD 40249.398564815

CURRENT ORBIT ELEMENTS IN APPARENT SIDEREAL CARTESIAN COORDINATES

POSITION(METERS)	-5839521.600	-3287664.500	4071553.100
VELOCITY(METERS/SEC)	1791.014500	3591.491500	5818.089900

STATION	DATE	TIME(UT)	CORRECTED RANGE	UNCERTAINTY	MISCLOSURE
4082	28 JAN 69	9 34 21.7000	2505094.64	2.00	-45.68
4082	28 JAN 69	9 34 41.6998	2488615.81	2.00	-53.92
4061	28 JAN 69	9 34 16.2000	2381823.98	2.00	-109.62
4061	28 JAN 69	9 34 36.1998	2480479.17	2.00	-113.16
4082	28 JAN 69	9 42 21.8000	3651357.54	2.00	-169.74
4760	28 JAN 69	9 42 11.1885	3351769.98	2.00	-282.57
4760	28 JAN 69	9 42 31.1877	3464788.52	2.00	-282.17
4760	28 JAN 69	9 42 51.1876	3578380.17	2.00	-283.08

PARTIAL UNCERTAINTIES OF ORBIT UNKNOWNS-FROM DDN(INV)

0.346249D 01 0.124982D 01 0.831489D 00 0.120211D-01 0.494939D-02 0.248116D-02

WEIGHTED SUM OF SQUARES OF MISCLOSURES = 414757.488
NUMBER OF OBSERVATIONS = 70
RMS MISCLOSURE = 76.975

Figure 20.

passes if the RMS misclosure is high. However, it is best to wait at least until the second iteration before removing data.

After the listing of a complete iteration, there are several items printed, these particular items being identical to that of a geometric adjustment. These are the Analysis of Misclosures by stations, guide matrix, normal equations, observations on each line, constraints, and the summary by observed lines. Samples of these can be seen in Figures 6, 15, 16, 17 and 18. As always, the analyst can repress the printing of some of these by using the proper PCODE.

After all of the above items have been printed (or repressed, as the case may be), there is a listing of corrections to orbit and error model unknowns for each pass (see Figure 21). This is extrememly valuable, especially after the second iteration, to determine the quality of the orbits. This information should be used in conjunction with the listing shown in Figure 20, where at the bottom are given the uncertainties of these particular orbital elements.

After the corrections to orbit and error model unknowns are listed come the results of the adjustment, which will be described in Section 5.4. After that the next iteration begins (if there is a next iteration) and everything is repeated.

5.4 The Output of Adjusted Coordinated and Related Information.

The most important part of any adjustment is the adjusted coordinates of the parameters and the standard deviations of these adjusted coordinates. The printed output from this program gives all this information and more for each station in the network.

Prior to printing out the information for each station, a short tabulation is given listing degrees of freedom, V'PV, σ_0^2, and σ_0. These values refer to the situation after the constraints have been added to the normal

CORRECTIONS TO ORBIT AND ERROR MODEL UNKNOWNS

```
  2     PASS 2
CORRECTION VECTOR        EPOCH= 28 JAN  69  9H 33M 56.0000S UT=MJD  40249.398564815
-143.67340819    -21.05024861    -166.81242405 -0.06976902  -0.03690959  -0.04738683

   UPDATED ORBIT ELEMENTS IN APPARENT SIDEREAL CARTESIAN COORDINATES
    POSITION(METERS)            -5839665.273     -3287685.550     4071386.288
    VELOCITY(METERS/SEC)         1790.955731      3591.454590     5818.042513


  3     PASS 3
CORRECTION VECTOR        EPOCH= 28 JAN  69 11H 23M 56.0000S UT=MJD  40249.474953704
-122.31373575    -88.52634812    -174.57801664  0.00583718  -0.04677484  -0.12510625

   UPDATED ORBIT ELEMENTS IN APPARENT SIDEREAL CARTESIAN COORDINATES
    POSITION(METERS)            -6031421.514     -3762718.626     3248854.522
    VELOCITY(METERS/SEC)         1111.462937      3194.186825     6233.512894


  5     PASS 5
CORRECTION VECTOR        EPOCH= 28 JAN  69 23H 18M 56.0000S UT=MJD  40249.971481481
  99.79297248     98.72556723    -193.35861013  0.02146740  -0.07336664   0.14568621

   UPDATED ORBIT ELEMENTS IN APPARENT SIDEREAL CARTESIAN COORDINATES
    POSITION(METERS)             4614260.193      5197589.926     3550155.041
    VELOCITY(METERS/SEC)         3545.638267       725.030533    -6111.083214


   UPDATED COORDINATES OF THE CENTER OF MASS
         -311.854          191.895          -563.219
```

Figure 21.

equations and the normal equations solved for the corrections to the parameters. A sample output is shown in Figure 22.

The output of the adjusted coordinates of the parameters is identical for every type of adjustment, a sample of which is shown in Figure 23. As can be seen, the results are self-explanatory. The adjusted coordinates and standard deviations are given in both the Cartesian coordinate system and in the φ, λ, h system with respect to the datum of adjustment. It should be mentioned here that the standard deviations are derived from the variance-covariance matrix multiplied by the value of σ_0^2.

Additional information printed is the direction of eigenvectors and square roots of eigenvalues of variance-covariance matrix. This may or may not be useful to the analyst. Another group of information is the off-diagonal elements of the weight-coefficient matrix as well as the correlation coefficients.

A printout is made for each station in the network. After all stations are output, the program will return for another iteration, if there is to be another iteration. If this is the last iteration, the words 'NORMAL TERMINATION' will be printed as the last line of output.

NUMBER OF DEGREES OF FREEDOM =	271
QUADRATIC SUM OF ALL THE RESIDUALS (VPV) =	266.5961
VARIANCE OF UNIT WEIGHT =	0.9837
STANDARD DEVIATION OF UNIT WEIGHT =	0.9918

Figure 22.

STATION NUMBER - 5410 MIDWAY SEC GEOC ELLIPSOID 7 SAO 1969

	X	Y	Z	LAT.	LONG.(+E)	ELL. HT.
PREL. COORD. -	-5618729.5352	-258198.3722	2997233.8079	28 12 43.2370	182 37 51.8560	-9.2900
CORRECTIONS -	-9.1203	-12.2942	11.6966	0 0 0.1862	0 0 0.4350	14.0551
ADJ. COORD. -	-5618738.6555	-258210.6664	2997245.5046	28 12 43.4232	182 37 52.2910	4.7651

VARIANCE-COVARIANCE MATRIX OF THE STATION POSITION

	91.331782	40.498980	-102.881694	0.052232	0.044362	2.647202
	40.498980	65.861489	-68.062532	0.044362	0.083648	2.362280
	-102.881694	-68.062532	150.026834	2.647202	2.362280	195.520361
STAND. DEV. -	9.5568	8.1155	12.2485	0.2285	0.2892	13.9829

DIRECTIONS OF EIGENVECTORS AND SQUARE ROOTS OF EIGENVALUES OF VARIANCE-COVARIANCE MATRIX -

LATITUDE	LONGITUDE	ELEVATION	AZIMUTH	AXIS LENGTH
48 21 31.2589	-145 27 21.5953	58 14 44.4656	41 52 9.7999	16.1149
5 5 57.0727	118 47 2.7514	25 23 49.2719	- 98 13 26.8569	6.0566
41 11 6.4421	24 18 21.8179	- 17 45 28.8403	- 16 58 11.2579	3.2937

3X3 WEIGHT COEFFICIENT MATRICES / CORRELATION COEFFICIENTS

	X	Y	Z	X	Y	Z
STA. NO. 5410 WITH STA. NO. 5410						
	92.840499	41.167985	-104.581205	1.000000	0.522177	-0.878906
	41.167985	66.949460	-69.186862	0.522177	1.000000	-0.684712
	-104.581205	-69.186862	152.505139	-0.878906	-0.684712	1.000000
STA. NO. 5410 WITH STA. NO. 5411						
	-46.725036	-39.920902	-71.530350	-0.797291	-0.478285	-0.700819
	-0.624932	17.405108	-74.699018	-0.012557	0.245560	-0.861838
	35.241776	5.129640	114.152416	0.469193	0.047951	0.872624

6. ADDITIONAL FEATURES

The first five chapters of this report give the details of using the OSUGOP program. Although it has not been mentioned, it is possible to make changes to any or all of the subprograms that will not alter the subprograms on the disk. There have been many occasions where there was a need to modify certain parts of OSUGOP to perform a special type of adjustment. This is done by including the source version of the subprograms (with the appropriate changes made) as part of the deck setup. This requires a completely different set of JCL cards (see Appendix III). There have been other occasions where additional programs have been written to perform certain required tasks. These programs are run separately from OSUGOP, but the output can be used by OSUGOP.

Although there have been many modifications used, probably the most important is the ability to read more than one set of normal equations and to perform an adjustment using all normal equations. Another very real problem is the ability to input correlated observations. The following is a brief description of the ways to handle these problems.

6.1 Addition of Normal Equations.

It was mentioned in section 3.2.3.1 that different systems of normal equations can be combined, and a description was given as to how they should be combined. At OSU there are two different techniques for adding normal equations. The one is a modification to the subroutine RDSOLN and the other is a separate program.

The modification to RDSOLN is really just the addition of a DO LOOP that causes the program to keep reading normal equations. The deck setup is as shown in Figure 4 except that the degrees of freedom and $V'PV$, punched normal equations, and extra 999 card are repeated for each set of normal equations. At the end of the last set of normal equations the E card is inserted.

The separate program that adds normal equations has been called ADDITION. It adds normal equations together and then prints and punches the combined normal equations. In addition to this, different weights can be applied to the different sets of equations.

6.2 Forming Normal Equations Using Correlated Observations.

The OSUGOP program was written to form normal equations from uncorrelated observations. However, the observations from the NOAA BC-4 World-wide Network are in the form of Greenwich Hour Angle and Declination for up to seven ficticious images from each camera plate. These observations are the result of a polynomial fit, and there is a full variance-covariance matrix for all of the observations. In order to use this data, which was recorded on 17 magnetic tapes, a special program was assembled to read these tapes and form the normal equations. The normal equations are compatible to those described earlier in this report and OSUGOP was used to perform the solution from the punched normal equations. This program is described in [Mueller, et al., in press].

REFERENCES

Blaha, Georges (1971). "Inner Adjustment Constraints with Emphasis on Range Observations." Reports of the Department of Geodetic Science, No. 148, The Ohio State University, Columbus.

Hartwell, J. G. and T. R. Lewis (1967). Integration of Orbits and Concomitant Variational Equations by Recurrent Power Series. D. Brown Associates, Inc., Melbourne, Florida.

Krakiwsky, Edward J and Allen J. Pope (1967). "Least Squares Adjustment of Satellite Observations for Simultaneous Directions or Ranges, Part 1 of 3: Formulation of Equations." Reports of the Department of Geodetic Science, No. 86, The Ohio State University, Columbus.

Krakiwsky, Edward J., J. Ferrier, and G. Blaha (1968). "Least Squares Adjustment of Satellite Observations for Simultaneous Direction or Ranges, Part 2 of 3: Computer Programs." Reports of the Department of Geodetic Science, No. 87, The Ohio State University, Columbus.

Krakiwsky, Edward J., J. Ferrier and J. P. Reilly (1967). "Least Squares Adjustment of Satellite Observations for Simultaneous Direction or Ranges, Part 3 of 3: Subroutines." Reports of the Department of Geodetic Science, No. 88, The Ohio State University, Columbus.

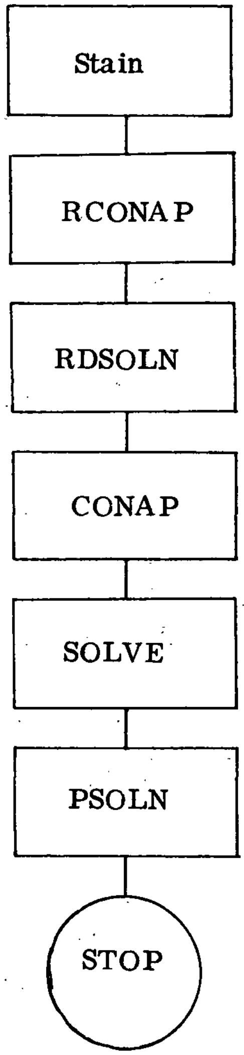
Mueller, Ivan I. (1969). Spherical and Practical Astronomy as Applied to Geodesy. Frederick Ungar Publishing Co., New York.

Mueller, Ivan I., J. P. Reilly, C. Schwarz and G. Blaha (1970). "SECOR Observations in the Pacific." Reports of the Department of Geodetic Science, No. 140, The Ohio State University, Columbus.

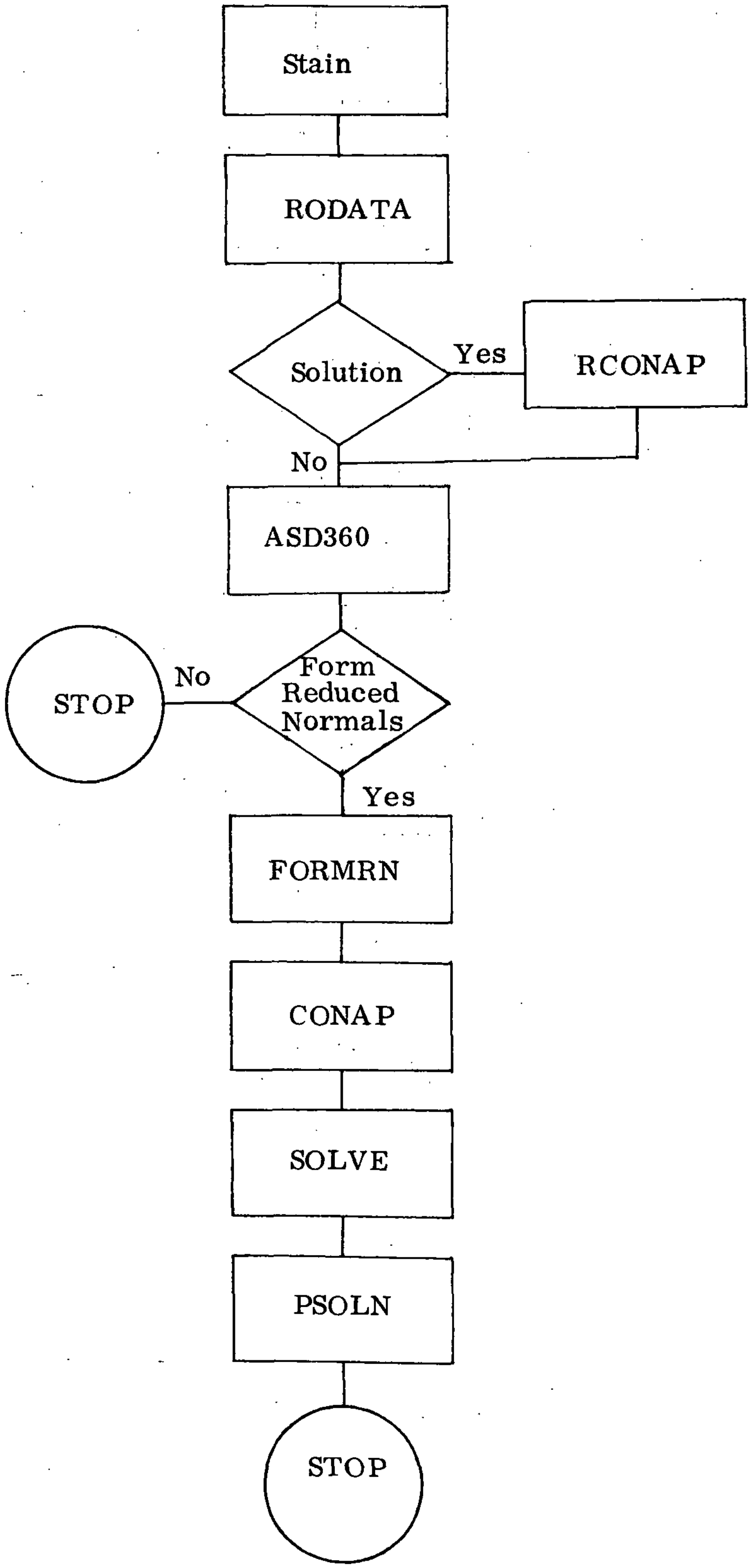
Mueller, Ivan I., M. Kumar, J. P. Reilly, and N. K. Saxena. (1973). "Analysis of the DOC/DOD Cooperative Worldwide Geodetic Satellite (BC-4) Network." Reports of the Department of Geodetic Science, No. 193, The Ohio State University, Columbus. (in press).

APPENDIX I

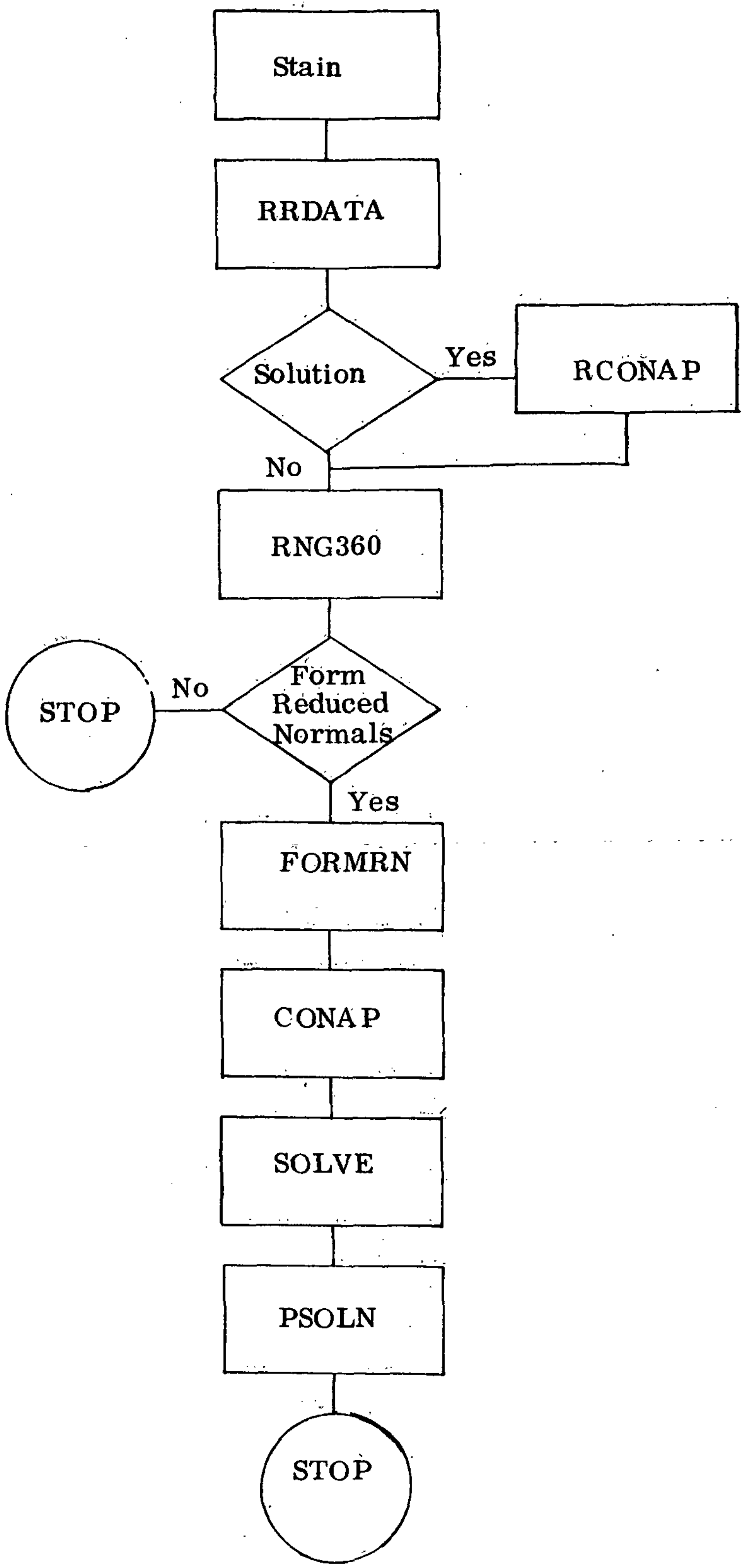
Flow Diagrams



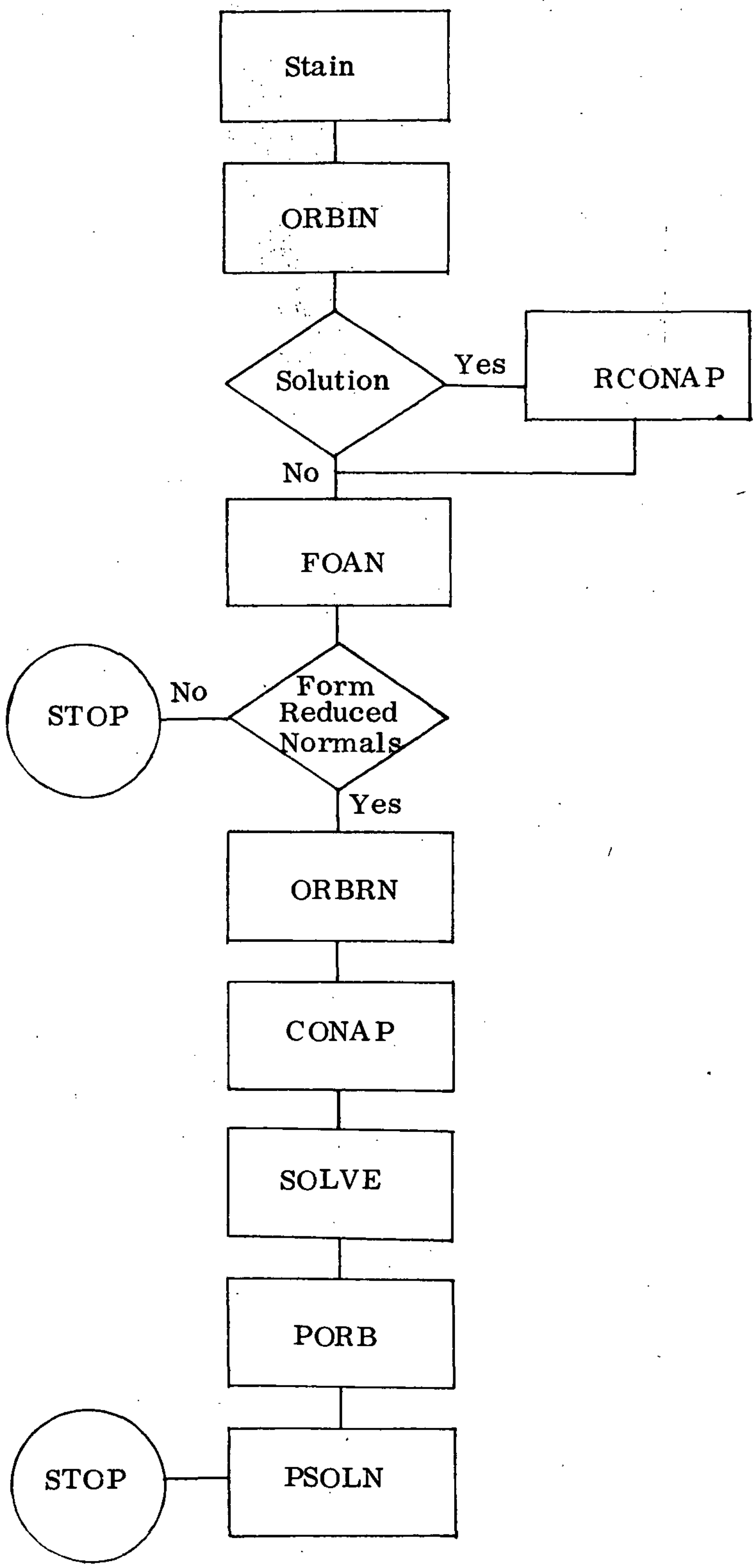
SOLUTION ONLY.



OPTICAL DATA, GEOMETRIC MODE.



RANGE DATA, GEOMETRIC MODE.



RANGE DATA, OPTICAL MODE.

APPENDIX II

Brief Description of Subprograms

Subprogram Listing

This is a listing of all subroutines and functions used in the OSUGOP program and a brief description of what is done by each. If the subprogram is a function, the letter F will appear in parenthesis after the name.

Name	Purpose
ANRADD (F)	Converts degrees, minutes, seconds into radians.
ASD 360	Processes optical directions. Reads data from disk, calls for the computation of satellite position, computes contribution to normal equations and contribution to V´PV.
CLEAR	Fills an array with Floating point zeros.
CONAP	Processes constraints on, and between, stations. Reads constraint codes, edits for wrong codes, calls CONAP1 and CONAP2.
CONAP1	Processes weighted constraints, adds contribution to normals, V´PV, and d.o.f.
CONAP2	Processes absolute constraints, adds contribution to normals, V´PV and d.o.f.
DANG	Converts radians to degrees, minutes, seconds.
DEDIT	Edits optical data based on preliminary station positions and deletes bad observations and bad events, based on the test distance criteria.
DELL	Used to compute ΔX, ΔY, ΔZ. Also propogates the error from the cartesian coordinates to the Geodetic coordinates.
DPDOT (F)	Takes the dot product of two vectors.

DRIVER	Acts as the driving program for Orbit integration.
EXPAND	Expansion of Power Series coefficients, used for orbital runs.
FOAN	Forms normal equations for short arc mode processing.
FORMRN	Forms reduced normals for geometric mode processing.
GSTD (F)	Computation of Greenwich Sidereal Time
KEPEQ	Solves Kepler's Equation
ICEPTCE	Converts from Keplerian to Cartesian Orbit Elements.
KSID2 (F)	Stops the program if an observation is from a station not included in the list of input stations.
KSTAID (F)	Searches table of station identifiers for the internal number of a station.
MAIN	This is the driving program. Everything starts and stops here.
MATRUP	Updates the matrizant with respect to time. Used for orbital runs.
MJD(F)	Computed Modified Julian Day.
ORBIN	Orbit input subroutine.
ORBIT	Orbit integration controller.
ORBRN	Forms reduced normals for short arc mode processing.
POLE	Computes polar motions values x and y.
PORB	Prints updated orbital elements and Error Model Terms.
PRENUT	Computes precession and Nutation.
PSOLN	Prints the solution.
RCONAP	Reads the constraint cards and writes constraint information on a disk.
RDSOLN	Reads normal equations punched on cards and sets up storage for a solution.

RNG 360	Processes range measurements. Accepts or rejects events based on test variance. Computes contribution to normal equations, $V'PV$ and d. o. f.
RODATA	Reads the optical data cards, rotates into terrestrial coordinate system, puts all data onto a disk.
ROT3	Performs an R_3 rotation to a vector.
RRDATA	Reads Range data input cards, puts all data onto a disk.
SATXYZ	Computes the satellite position from approximate station coordinates and three or more range measurements.
SOLVE	Solves Normal Equations and computes inverse.
STAIN	Reads station coordinates and datum information from cards.
SWITCH	Switches rows and columns in a matrix.
UPDATE	Evaluates position and velocity at time $t + \Delta t$, given the position and velocity at time t. Used for Orbital Solutions.
UVWD	Converts geodetic to rectangular coordinates.
UVWTG	Converts rectangular to geodetic coordinates.
UVWTG2	Same as UVWTG. It is located in a different overlay.
UVWTG3	Same as UVWTG and UVWTG2. It is located in a different overlay.
VARIEQ	Generates the power series required to evaluate the matrix solution of variational equations. Used for orbital solutions.

APPENDIX III

JCL
(Job Control Cards)

JCL

```
// (10000,1000),CLASS=C
//STEP1   EXEC   PROC=FORTRANG,TIME=(,20)
//CMP.SYSIN   DD   *


      INSERT SOURCE SUBPROGRAMS HERE


/*
//STEP2   EXEC PGM=IEWL,PARM='MAP,LIST,OVLY,ID',TIME=(0,20)
//MYLIB   DD   DSN=SCJ032.MUELLER,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(2,1))
//SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR
//        DD DSN=SYS2.FORTSSP,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSLMOD DD DSNAME=&GO(MAIN),UNIT=SYSDA,SPACE=(CYL,(1,1,1)),
//           DISP=(NEW,PASS)
//SYSLIN DD DSNAME=*.STEP1.CMP.SYSLIN,DISP=(OLD,DELETE)
//   DD *
  ALIAS        GEOMSG
  INCLUDE      MYLIB(GEOMSG)
  OVERLAY      ALPHA
  INSERT       OBSD,MJD,PRENUT,POLE,GSTD,STAPLH,DPDOT,ANRADD,UVWTG2
  INSERT       RCONAP,KSID2
  INSERT       STAIN,UVWD
  OVERLAY      GAMMA
  INSERT    RODATA,ASD360,DEDIT,DEDITC
  OVERLAY    GAMMA
  INSERT    RRDATA,RNG360,SATXYZ,RANGED
  OVERLAY      GAMMA
  INSERT       ERDCON,ORBCOM,ORBPAR,ORBIT,EXPAND,VARIEQ,UPDATE
  INSERT       CLEAR
  INSERT       ORBIN,KEPTCE,KEPEQ,ROT3
  INSERT       FOAN,DRIVER,MATRUP
  OVERLAY      ALPHA
  INSERT       NORMEQ
  OVERLAY      BETA
  INSERT       ORBRN
  OVERLAY    BETA
  INSERT    FORMRN
  OVERLAY    BETA
  INSERT    RDSOLN
  OVERLAY      BETA
```

JCL required when a subroutine (or subroutines) is different from that on the disk. This does not change the subroutines on the disk, but merely overrides them.

JCL (Continued)

```
  INSERT        CONAP
  OVERLAY       DELTA
  INSERT        CONAP1,UVWTG3
  OVERLAY       DELTA
  INSERT        CONAP2
  OVERLAY       BETA
  INSERT        SOLVE,SWITCH
  OVERLAY       ALPHA
  INSERT        PORB
  OVERLAY       ALPHA
  INSERT        PSOLN,DANG,UVWTG,DELL,DEIGEN
/*
//GO   EXEC PGM=*.STEP2.SYSLMOD,TIME=(04,20),REGION=252K
//FT01F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//                 DCB=(LRECL=400,BLKSIZE=404,RECFM=VS)
//FT02F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//                 DCB=(LRECL=404,BLKSIZE=412,RECFM=VS)
//FT03F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//                 DCB=(LRECL=400,BLKSIZE=404,RECFM=VS)
//FT04F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//                 DCB=(LRECL=400,BLKSIZE=404,RECFM=VS)
//FT06F001 DD SYSOUT=A
//FT07F001 DD SYSOUT=B
//FT05F001  DD *


          INSERT DATA HERE


/*
//
```

```
//   (2000,200),CLASS=B
//JOBLIB  DD DSNAME=SCJ032.MUELLER,DISP=SHR,PARM='ID'
//GO EXEC PGM=OSUGOP,TIME=2
//FT01F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//               DCB=(LRECL=400,BLKSIZE=404,RECFM=VS)
//FT02F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//               DCB=(LRECL=408,BLKSIZE=412,RECFM=VS)
//FT03F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//               DCB=(LRECL=400,BLKSIZE=404,RECFM=VS)
//FT04F001      DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE),
//               DCB=(LRECL=400,BLKSIZE=404,RECFM=VS)
//FT06F001 DD SYSOUT=A
//FT07F001 DD SYSOUT=B
//FT05F001  DD *


          INSERT DATA HERE


/*
//
```

JCL required to use the standard program on the disk.

APPENDIX IV

Fortran IV Program with Subroutines

```
C
C              PROBLEM  CODE  DEFINITIONS
C
C    COLUMN    MEANING
C       1. OVERALL PROBLEM CODE
C          PCODE(1)=1  MEANS  OPTICAL  PROGRAM
C                   2  MEANS  RANGE
C                   3  MEANS  SOLUTION ONLY RUN
C                   4  MEANS ORBITAL MODE, OPTICAL OBSERVATIONS
C                   5  MEANS ORBITAL MODE, RANGE OBSERVATIONS
C                   6  MEANS ORBITAL MODE, MIXED OBSERVATIONS
C          PCODE(1)=7  MEANS OPTICAL PROGRAM, GEOMETRIC MODE(GEOS FORMAT)
C       2. PERFORM SOLUTION?
C          PCODE(2)=1  MEANS YES
C                   0  MEANS NO
C          PCODE(1)=3 IMPLIES PCODE(2)=1
C       3. MAXIMUM NUMBER OF ITERATIONS?
C          PCODE(1) MUST EQUAL 1 OR 2,
C          PCODE(2) MUST EQUAL 1,
C          PCODE(5) MUST EQUAL 1,  FOR ONE OR MORE COMPLETE ITERATIONS
C       5. FORM NORMALS?
C
C              PROCESSING CODES
C                                        1 MEANS YES,  0 MEANS NO
C       6. SIMULATE GUIDE MATRIX?
C       7. PRINT NORMALS?
C       8. PERFORM  SUMMARY BY OBSERVED LINES?
C       9. PUNCH NORMALS IN ASD FORMAT?
C      10. SUMMARIZE RESULTS
C            PCODE(10)=0 DO NOT PRINT SUMMARY
C                     =1 PRINT THE DX'S AND STANDARD DEVIATIONS
C                     =2 PRINTS THE X,Y,Z'S AND STANDARD DEVIATIONS
C                     =3 PRINTS THE LATITUDE,LONGITUDE AND HEIGHT
C                     =4 PRINTS BOTH X,Y,Z & LAT.,LONG, & H
C      11. PRINT SATELLITE POSITION FOR EACH EVENT?
C            0 MEANS NO
C            1 MEANS PRINT XYZ AND GEODETIC COORDINATES
C            2 MEANS PRINT XYZ ONLY
C            3 MEANS PRINT GEODETIC COORDINATES ONLY
C      12. THIS PARAMETER DESCRIBES WHERE THE STANDARD DEVIATIONS OF THE
C            INDIVIDUAL OBSERVATIONS (USED TO FORM THE WEIGHTS) ARE TO BE FOUND
C            PCODE(12)=0 MEANS TO READ THE OBSERVATIONAL STANDARD DEVIATION
C            FROM THE CARD CONTAINING THE OBSERVATION.
C            PCODE(12)=1 MEANS TO ASSOCIATE A SINGLE STANDARD DEVIATION WITH
C            ALL OBSERVATIONS FROM A GIVEN STATION.** THE STANDARD DEVIATIONS
C            TO BE ASSOCIATED WITH EACH STATION ARE GIVEN IN COLUMNS 73-79 OF
C            THE CARD CONTAINING THE INPUT COORDINATES OF THE STATION.
C            PCODE(12)=2 MEANS TO ASSOCIATE A SINGLE STANDARD DEVIATION WITH
C            ALL OBSERVATIONS.** THIS NUMBER IS FOUND IN COLS. 21-30 OF THE
C            CARD CONTAINING THE TEST DISTANCE (OPTICAL) OR TEST VARIANCE
C            (RANGE).
C            ** IN THE CASE OF OPTICAL OBSERVARTIONS, THIS NUMBER IS INTERPRETE
```

```
C             AS THE STANDARD DEVIAION OF THE DECLINATION AND OF THE RIGHT
C             ASCENSION TIMES THE COSINE OF THE DECLINATION, AND THE
C             COVARIANCE IS SET TO ZERO.
C      CODES WHICH APPLY TO ORBITAL MODE PROCESSING ONLY
C        14. TREAT COORDINATES OF CENTER OF MASS AS UNKNOWNS? (ORBITAL MODE 653Y'
C        15. PUNCH UPDATED ORBIT ELEMENTS?  (ORBITAL MODE ONLY)
C
C
C       SOLUTION CODES
C
C        16. WRITE NORMALS AND INVERSE DURING SOLUTION PROCESSING?
C              0 MEANS PRINT NOTHING
C              1 MEANS PRINT PIVOT ELEMENTS
C              2 MEANS ALSO PRINT NORMALS AND INVERSE
C              3 MEANS ALSO PRINT REARRANGED NROMALS AND INVERSE
C        17. PUNCH ADJUSTED STATION XYZ AND VARIANCES FOR INPUT TO BADEKAS'
C              DATUM TRANSFORMATION PROGRAM?
C        18. PUNCH ADJUSTED STATION POSITIONS?
C        19. COMPUTE EIGENVECTORS OF VARIANCE-COVARIANCE MATRIX
C        20. COMPUTE CORRELATION COEFFICIENTS
C
      COMMON/NSTA/NSTA
      INTEGER*2 ENDSIG/1HE/,CONTIN
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      REAL*8 TITLE(10)
    3 CONTINUE
      WRITE(6,6001)
 6001 FORMAT(1H1,20(/))
    4 READ(5,5001) TITLE,CONTIN
 5001 FORMAT(9A8,A7,A1)
      IF(CONTIN.EQ.ENDSIG) GO TO 5
      WRITE(6,6012) TITLE
 6012 FORMAT(30X,9A8,A7)
      GO TO 4
    5 CONTINUE
C
      READ(5,5050) PCODE
 5050 FORMAT(80I1)
      WRITE(6,6050) PCODE
 6050 FORMAT(////10X,'PROBLEM CODES',10X,20I1)
      CALL STAIN
      J=PCODE(1)
      GO TO (100,200,300,400,500,600,100),JCODE
  100 CALL RODATA
      GO TO 105
  200 CALL RRDATA
      GO TO 105
  500 CALL ORBIN
      GO TO 105
  105 IF(PCODE(2).EQ.1) CALL RCOMAP
      NITR=PCODE(3)
```

```
      DO 110 I=1,NITR
      WRITE(6,6010) I
 6010 FORMAT(1H1///25X,'BEGIN ITERATION ',I5)
C              RUN OPTICAL
      IF(JCODE.EQ.1.OR.JCODE.EQ.7) CALL ASD360
C              RUN RANGE PROGRAM
      IF(JCODE.EQ.2) CALL RNG360
      IF(JCODE.EQ.5) CALL EOAM
      IF(PCODE(5).NE.1) GO TO 890
      IF(JCODE.LT.3.OR.JCODE.EQ.7) CALL FORMNM
      IF(JCODE.GT.3.AND.JCODE.LT.7) CALL ORBEN
      ASSIGN 110 TO JRTN
      GO TO 800
  110 CONTINUE
      GO TO890
  300 CONTINUE
      CALL RCONAP
      CALL RDSOLN
      ASSIGN890 TO JRTN
      GO TO 800
  800 CONTINUE
      CALL CONAP
      CALL SOLVE
      IF(JCODE.GT.3.AND.JCODE.LT.7) CALL PORB
      CALL PSOLN
      GO TO JRTN,(110,890)
C
  600 CONTINUE
  400 CONTINUE
  890 CONTINUE
      WRITE(6,6002)
 6002 FORMAT(//'ONORMAL TERMINATION'/1H1)
      STOP
      END
```

```
      DOUBLE PRECISION FUNCTION DPDOT(X,Y,N)
      DOUBLE PRECISION X(N),Y(N)
      DPDOT=0.0
      DO 10 I=1,N
   10 DPDOT=DPDOT+X(I)*Y(I)
      RETURN
      END
```

```
      SUBROUTINE FORMRN
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/NSTA/NSTA
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      COMMON/WPW/WPW,XPU,IDEGF,IFSTA
      DIMENSION DDN(3,3),DDK(3),L1(3),L2(3),BNDDNI(3,3),TN(3,3),TK(3)
      INTEGER*2 L,LSOLVE
      INTEGER CONTIN,ENDSIG/1HE/
      COMMON/STAORD/KORDER(150)
      COMMON/NORMEQ/REDN(3,3,820),U(3,40),L(820),LSOLVE
      DIMENSION BN(3,3,40),LG(40)
C  FORM REDUCED NORMAL EQUATIONS FOR UP TO 40 STATIONS
      DIMENSION KSTATE(40)
      LOC(K)=(K*(K+1))/2
      MAXSTA=40
      IF(NSTA.GT.MAXSTA) GO TO 901
C
C   THE REDUCED NORMAL EQUATIONS ARE STORED AS 3 X 3 BLCCKS IN THE ARRAY REDN.
C   ONLY THE UPPER TRIANGULAR PART OF THE REDUCED NORMAL EQUATIONS IS STORED.
C     THE BLOCKS OF THE REDUCED NORMAL EQUATIONS ARE NUMBERED
C       ACCORDING TO THE FOLLOWING SCHEME:
C
C         1    2    4    7    11
C              3    5    8    12
C                   6    9    13
C                        10   14
C                             15            ET CETERA
C
C  L(820) IS THE GUIDE MATRIX
C  L=1  SIGNIFIES A NON ZERO BLOCK
C  L=0  SIGNIFIES A ZERO BLOCK
      IB=LOC(NSTA)
      DO 100 JB=1,IB
      DO 99 I=1,3
      DO 99 J=1,3
   99 REDN(I,J,JB)=0.0
  100 L(JB)=0
C
      BACKSPACE 2
      READ(2)      (((BN(I,J,KSTA),I=1,3),U(J,KSTA),J=1,3),
     XKSTA=1,NSTA)
      REWIND 2
C
C  STASH  DIAGONAL BLOCKS
      DO 110 KSTA=1,NSTA
      IB   =LOC(KSTA)
      DO 108 I=1,3
      DO 108 J=1,3
  108 REDN(I,J,IB)=BN(I,J,KSTA)
  110 CONTINUE
C
      FDEGF=IDEGF
      IF(PCODE(9).EQ.1) WRITE(7,7010) FDEGF,WPW
 7010 FORMAT(16X,2F16.6)
C   READ BLOCKS FROM EACH EVENT AND REDUCE NORMAL EQUATIONS
C
  150 READ(2)          NSTE,DDN,DDK,(((BN(I,J,IS),I=1,3),J=1,3),
```

```
     1KSTATE(IS),IS=1,NSTE),CONTIN
C
      DO 180 IS=1,NSTE
      ISTA=KSTATE(IS)
      IB=ISTA
      CALL DGMPRD(BN(1,1,IS),DDN,BNDDNI,3,3,3)
      CALL DGMPRD(BNDDNI,DDK,TK,3,3,1)
      DO 155 I=1,3
  155 U(I,ISTA)=U(I,ISTA)-TK(I)
      DO 180 JS=1,NSTE
      JSTA=KSTATE(JS)
      JB=JSTA
C   SKIP IF (ISTA.GT.JSTA), SINCE ONLY THE UPPER TRIANGULAR PART OF THE
C       REDUCED NORMAL EQUATIONS IS BEING COMPUTED AND SAVED.
      IF(ISTA.GT.JSTA) GO TO 180
C   (IB,JB) GIVES THE ROW AND COLUMN NUMBER OF THE BLOCK IN THE REDUCED
C       NORMAL EQUATIONS CURRENTLY BEING PROCESSED.
C
C  SET INDICATOR
      NB=LOC(JB-1)
      NB=IB+NB
      L(NB)=L(NB)+1
C  PERFORM REDUCTION
      CALL DGMPRD(BNDDNI,BN(1,1,JS),TN,3,3,3)
      DO 130 I=1,3
      DO 130 J=1,3
  130 REDN(I,J,NB)=REDN(I,J,NB)-TN(I,J)
  180 CONTINUE
C   IF END OF DATA, GO OUT OF LOOP
      IF(CONTIN.EQ.ENDSIG) GO TO 400
C   IF NOT, RETURN TO PROCESS ANOTHER EVENT
      GO TO 150
C
C   ENTER HERE WHEN ALL EVENTS HAVE BEEN PROCESSED.
  400 CONTINUE
C
C  SIMULATE KRAKIWSKI'S GUIDE MATRIX
      IF(PCODE(6).NE.1) GO TO 441
C
      WRITE(6,6001)
 6001 FORMAT(1H1,10(/),20X,'GUIDE MATRIX')
      DO 440 ISTA=1,NSTA
      IB=0
      LG(1)=1000
      DO 435 JSTA=ISTA,NSTA
      JB=LOC(JSTA-1)+ISTA
      IF(L(JB).EQ.0) GO TO 435
      IB=IB+1
      LG(IB)=KORDER(JSTA)
  435 CONTINUE
C
      IB=IB+1
      IF(IB.GT.1) LG(IB)=999
  439 WRITE(6,6002) KORDER(ISTA),(LG(I),I=1,IB)
 6002 FORMAT(20X,I5,5X,18I5,200(/30X,18I5))
  440 CONTINUE
  441 CONTINUE
C
```

```
C  PRINT NORMALS IN ASD FORMAT,AND PUNCH IF DESIRED.
      WRITE(6,6003)
 6003 FORMAT(1H1//'                NORMAL EQUATIONS (SEE GUIDE MATRIX)'//)
      DO 450 ISTA=1,NSTA
      DO 442 I=1,3
  442 DDK(I)=-U(I,ISTA)
      IB=0
      JB=LOC(ISTA)
      IF(L(JB).GT.0) IB=1
C   PUNCH NORMALS
      IF(PCODE(9).NE.1) GO TO 443
      WRITE(7,7001) KORDER(ISTA)
 7001 FORMAT(14I5)
      WRITE(7,7006) DDK
 7006 FORMAT(3(F16.10,5X))
      WRITE(7,7008) ((REDN(I,J,JB),J=1,3),I=1,3)
 7008 FORMAT(3F16.10/3F16.10/3F16.10)
C
  443 CONTINUE
C   PRINT DIAGONAL BLOCK
      IF(PCODE(7).NE.1) GO TO 444
      WRITE(6,6004) KORDER(ISTA)
 6004 FORMAT(//I5)
      WRITE(6,6006) DDK
 6006 FORMAT(/3(F16.10,5X))
      WRITE(6,6008) ((REDN(I,J,JB),J=1,3),I=1,3)
 6008 FORMAT(3F16.10)
  444 CONTINUE
C   PRINT OFF-DIAGONAL BLOCKS
      KSTA=ISTA+1
      IF(ISTA.EQ.NSTA) GO TO 448
      DO 445 JSTA=KSTA,NSTA
      JB=LOC(JSTA-1)+ISTA
      IF(L(JB).EQ.0) GO TO 445
      IB=IB+1
      IF(PCODE(9).NE.1) GO TO 7445
      WRITE(7,7001) KORDER(JSTA)
      WRITE(7,7008) ((REDN(I,J,JB),J=1,3),I=1,3)
 7445 CONTINUE
      IF(PCODE(7).NE.1) GO TO 445
      WRITE(6,6004) KORDER(JSTA)
      WRITE(6,6008) ((REDN(I,J,JB),J=1,3),I=1,3)
  445 CONTINUE
  448 I=1000
      IF(IB.GT.0) I=999
      IF(PCODE(7).EQ.1) WRITE(6,6004) I
      IF(PCODE(9).EQ.1) WRITE(7,7001) I
  450 CONTINUE
      IF(PCODE(8).NE.1) GO TO 478
      WRITE(6,6010)
 6010 FORMAT(10(/),20X,'OBSERVATIONS ON EACH LINE')
      IB=NSTA-1
      DO 475 ISTA=1,IB
      KSTA=ISTA+1
      DO 475 JSTA=KSTA,NSTA
      WRITE(6,6011) KORDER(ISTA),KORDER(JSTA),L(LOC(JSTA-1)+ISTA)
 6011 FORMAT(8I10)
  475 CONTINUE
```

```
 478 CONTINUE
     RETURN
 901 CONTINUE
     WRITE(6,9001) MAXSTA,NSTA
9001 FORMAT('    FORMRN IS PRESENTLY DIMENSIONED TO HANDLE ONLY',I5,
    1'   UNKNOWN STATIONS.'/20X,' THIS PROBLEM HAS',I5,'  UNKNOWN STATI
    2IONS.'/10X,'EXECUTION IS TERMINATED BY PROGRAM.')
     STOP
     END
```

```
      SUBROUTINE RDSOLN
C  THIS SUBROUTINE READS THE NORMAL EQUATION FROM CARDS AND SETS UP
C      STORAGE FOR A SOLUTION,SIMULATING STORAGE AFTER EXECUTION OF FORMRN.
C  THIS SUBROUTINE IS CALLED ONLY FOR A SOLUTION-ONLY RUN.
C  THE  INPUT IS COMPATABLE WITH KRAKIWSKI'S  SOLUTION PROGRAM.
C  THE  STATION COORDINATES  MUST BE INITIALIZED BY CALLING STAIN.
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 L,LSOLVE
      INTEGER ENDSIG/1HE/,CONTIN
      COMMON/NSTA/NSTA/STAORD/KORDER(150)/WPW/WPW,XPU,IDEGF,IFSTA
      COMMON/NORMEQ/REDN(3,3,820),U(3,40),L(820),LSOLVE
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      LOC(K)=(K*(K+1))/2
C
      NB=LOC(NSTA)
      DO 10 IB=1,NB
      L(IB)=0
      DO 10 I=1,3
      DO 10 J=1,3
   10 REDN(I,J,IB)=0.0
      DO 12 IB=1,NSTA
      DO 12 I=1,3
   12 U(I,IB)=0.0
C
      READ(5,5001) VUW,FDEGF,WPW
 5001 FORMAT(3D16.8)
      IDEGF=FDEGF
C
C  READ NORMALS AND STORE
  100 READ(5,5002) ID,CONTIN
 5002 FORMAT(I5,74X,A1)
      IF(ID.EQ.999) CONTIN=ENDSIG
      IF(CONTIN.EQ.ENDSIG) GO TO 600
      ISTA=KSTAID(ID)
      IF(ISTA.LE.0) GO TO 900
C     READ CONSTANT COLUMN
      READ(5,5003) (U(I,ISTA),I=1,3)
C  SWITCH SIGN
      DO 110 I=1,3
  110 U(I,ISTA)=-U(I,ISTA)
C  READ DIAGONAL BLOCK
      NB=LOC(ISTA)
      READ(5,5004) ((REDN(I,J,NB),J=1,3),I=1,3)
 5003 FORMAT(3(D16.8,5X))
 5004 FORMAT(3D16.8)
      L(NB)=1
C
C  READ OFF-DIAGONAL BLOCK
  150 READ(5,5002) ID
      IF(ID.EQ.999) GO TO 100
      JSTA=KSTAID(ID)
      IF(JSTA.LE.0) GO TO 900
C   SWITCH SUBSCRIPTS IF NECESSARY SO THAT STORAGE IS MADE IN UPPER TRIANGULAR P
C     PART OF REDUCED NORMAL EQUATIONS.
      IF(JSTA.GE.ISTA) GO TO 160
      NB=LOC(ISTA-1)+JSTA
      READ(5,5004) ((REDN(J,I,NB),J=1,3),I=1,3)
```

```
      GO TO 170
  160 CONTINUE
      NB=LOC(JSTA-1)+ISTA
      READ(5,5004) ((REDN(I,J,NB),J=1,3),I=1,3)
  170 CONTINUE
 6001 FORMAT(3I5)
      L(NB)=1
      GO TO 150
C
  900 WRITE(6,6000) ID
 6000 FORMAT('OSTATION NUMBER NOT FOUND IN INPUT LIST',I6,' PROGRAM STOP
     1S.')
      STOP
C
  600 CONTINUE
      IF(PCODE(7).EQ.0) GO TO 620
      WRITE(6,6003)
 6003 FORMAT(///T30,'NORMAL EQUATIONS')
      DO 615 ISTA=1,NSTA
      WRITE(6,6002) KORDER(ISTA),ISTA
 6002 FORMAT(///3I10)
      WRITE(6,6004) (U(I,ISTA),I=1,3)
      DO 615 JSTA=ISTA,NSTA
      NB=LOC(JSTA-1)+ISTA
      WRITE(6,6002) KORDER(ISTA),KORDER(JSTA),NB
      WRITE(6,6004) ((REDN(I,J,NB),J=1,3),I=1,3)
 6004 FORMAT(3F16.10)
  615 CONTINUE
  620 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STAIN
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      INTEGER ENDSIG/1HE/,CONTIN
      COMMON/NSTA/NSTA
      COMMON/STAORD/KORDER(150)
      INTEGER STANAM,IDS*2
      INTEGER*2 PLUS/1H+/
      INTEGER*2 ISGNP,IPHID,IPHIM,LONGD,LONGM,ISGNL
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      DIMENSION UNCE(3)
C     UNCE FOR THE MOMENT IS A DUMMY ARRAY
      COMMON/STAPLH/STAPLH(2,150)
      COMMON/OBSD/OBSD(150),OVOBSD
      MAXSTA=150
      WRITE(6,6000)
 6000 FORMAT(1H1)
 6001 FORMAT(1H1,20(/))
      WRITE(6,6002)
 6002 FORMAT(/////4X,29HDATUMS INVOLVED IN ADJUSTMENT,//)
C  INPUT DATUMS
   10 READ(5,5002) IDD,AE,BE,CONTIN
 5002 FORMAT(I2,2F12.3,53X,A1)
      IF(CONTIN.EQ.ENDSIG) GO TO 30
      DATPRM(1,IDD)=AE
      DATPRM(2,IDD)=BE
      READ(5,5003)(DATNAM(I,IDD),I=1,4)
 5003 FORMAT(4A8)
      WRITE(6,6003) IDD,(DATNAM(I,IDD),I=1,4),(DATPRM(I,IDD),I=1,2)
 6003 FORMAT(6H0DATUM,I3,3X,4A8,3HA= ,F10.2,12H METERS  B= ,F10.2,
     17H METERS)
      GO TO 10
C
   30 CONTINUE
C         STATION INPUT
      WRITE(6,6005)
 6005 FORMAT(1H1///40X,29HINPUT COORDINATES OF STATIONS)
      KSTA=0
   35 KSTA=KSTA+1
      READ(5,5005)IDD,IDTS,(STANAM(I,KSTA),I=1,5),ISGNP,IPHID,IPHIM,PHIS
     1,LONGD,LONGM,FLONGS,H,UNCE,OBSD(KSTA),CONTIN
 5005 FORMAT(I4,I2,4A4,A2,A1,2(2I3,F8.4), F10.2,3F3.1,F7.2,A1)
      IF(CONTIN.EQ.ENDSIG) GO TO 50
      PHI=ANRADD(ISGNP,IPHID,IPHIM,PHIS)
      ISGNL=PLUS
      FLONG=ANRADD(ISGNL,LONGD,LONGM,FLONGS)
      KORDER(KSTA)=IDD
      IDS(KSTA)=IDTS
      STAPLH(1,KSTA)=PHI
      STAPLH(2,KSTA)=FLONG
      CALL UVWD(DATPRM(1,IDTS),DATPRM(2,IDTS),PHI,FLONG,H,STAUVW(1,KSTA)
     1,STAUVW(2,KSTA),STAUVW(3,KSTA))
      WRITE(6,6006)IDD,(STANAM(I,KSTA),I=1,5),IDTS,(DATNAM(I,IDTS),I=1,4
     1),ISGNP,IPHID,IPHIM,PHIS,ISGNL,LONGD,LONGM,FLONGS,H
 6006 FORMAT(1H0,I4,8X,4A4,A2,10X,5HDATUM,I4,4X,4A8/10X,20HGEODETIC COOR
     1DINATES,2(6X,A1,2I3,F8.4),F12.4)
```

```
      WRITE (6,6007) (STAUVW(I,KSTA),I=1,3)
 6007 FORMAT(10X,21HCARTESIAN COORDINATES,3F16.3)
      GO TO 35
   50 CONTINUE
      NSTA=KSTA-1
      NSTAUN=3*NSTA
      REWIND 3
      RETURN
      END
```

```
      SUBROUTINE ASD360
C   S/360 VERSION OF ASD PROGRAM FOR OPTICAL SATELLITE DIRECTIONS
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      INTEGER*4 ENDSIG/1HE/,CONTIN,DELCOD(2)/1H ,1H*/,ECODE
      INTEGER*2 PLUS/1H+/
      INTEGER*2 ISGNP,IPHID,IPHIM,LONGD,LONGM,ISGNL
      INTEGER*2 ID(50),KEY(50),IHR(50),MIN(50),IDAY(50),IYR(50),IRAH(50)
     1,IRAM(50),ISGND(50),IDECD(50),IDECM(50),IDAT(50,11)
      COMMON/DEDITC/ALFS(50),DEC(50),U(3,50),S(3),D(50),SDC(3,50),EVSUM,
     1GAST,STAXYZ(3,50),GQI,
     2TD,KSTATE(50),IPASS(50),NSTE,NSUSED,ECODE
      COMMON/NSTA/NSTA
      INTEGER STANAM,IDS*2
      DIMENSION  MONTH(50)
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/STAORD/KORDER(150)
      EQUIVALENCE(ID(1),IDAT(1,1)),(KEY(1),IDAT(1,2)),(IHR(1),IDAT(1,3))
     1,(MIN(1),IDAT(1,4)),(IDAY(1),IDAT(1,5)),(IYR(1),IDAT(1,6)),(IRAH(1
     2),IDAT(1,7)),(IRAM(1),IDAT(1,8)),(ISGND(1),IDAT(1,9)),(IDECD(1),ID
     3AT(1,10)),(IDECM(1),IDAT(1,11))
      DIMENSION SEC(50),RAS(50),DECS(50),VARRA(50),VARDEC(50),COVRAD(50)
     1,DAT(50,6)
      EQUIVALENCE(SEC(1),DAT(1,1)),(RAS(1),DAT(1,2)),(DECS(1),DAT(1,3)),
     1(VARRA(1),DAT(1,4)),(VARDEC(1),DAT(1,5)),(COVRAD(1),DAT(1,6))
      DIMENSION DN(3,3,150),BN(3,3,50),DDN(3,3),DK(3,150),DDK(3),A(2,3),
     1W(2,2),DL(2)
      DIMENSION PM(3,3),AP(2,3)
      DIMENSION L1(3),L2(3),TA(3)
      COMMON/WPW/WPW,XPU,IDEGF,NFSTA
      DIMENSION NOBSTA(150)
      REAL*4 VPVSTA(150)
      MAXSTE=50
      SPR=206264.80625D0
      PI=3.14159265358D0
      PI2=2.0*PI
      RPD=180.0/PI
      WPWSP=0.0
C
      REWIND 2
      REWIND 3
      READ(3) TD
      WRITE(6,6004) TD
 6004 FORMAT(//20X,'TEST DISTANCE =',F20.2,'     SECONDS OF ARC')
C
C    START DATA INPUT
      DO 70 KSTA=1,NSTA
      NOBSTA(KSTA)=0
      VPVSTA(KSTA)=0.0
      DO 70 I=1,3
      DK(I,KSTA)=0.0
      DO 70 J=1,3
      DN(I,J,KSTA)=0.0
   70 CONTINUE
      KEVENT=0
      EPR=0.0
```

```
  210 CONTINUE
      READ (3) IEVENT,NSTE,GAST,PM,EPR,
     1((IDAT(IS,J),J=1,11),MONTH(IS),(DAT(IS,J),J=1,6),ALFS(IS),DEC(IS),
     2KSTATE(IS),IS=1,NSTE),CONTIN
      DO 270 IS=1,NSTE
      KSTA=KSTATE(IS)
      CALL DGMPRD(PM,STAUVW(1,KSTA),STAXYZ(1,IS),3,3,1)
  270 CONTINUE
      WRITE(6,6008) IEVENT
 6008 FORMAT(/ 1X,'EVENT',I6)
C
      CALL DEDIT
C
      DO 280 IS=1,NSTE
  280 WRITE(6,6010) ID(IS),KEY(IS),IHR(IS),MIN(IS),SEC(IS),IDAY(IS),
     1MONTH(IS),IYR(IS),IRAH(IS),IRAM(IS),RAS(IS),ISGND(IS),IDECD(IS),
     2IDECM(IS),DECS(IS),VARRA(IS),VARDEC(IS),COVRAD(IS),D(IS),
     3DELCOD(IPASS(IS))
 6010 FORMAT(I7,A1,2I3,F9.5,3X,I3,A3,I2,2I3,F8.4,3X,A1,I2,I3,F8.4,
     15X,3F6.2,F10.1,2X,A1)
C
      IF(ECODE.GT.1) GO TO 630
      IF(PCODE(11)) 610,630,610
  610 IF(PCODE(11)-3) 611,612,611
  611 WRITE(6,6024) S
 6024 FORMAT('  SATELLITE POSITION',3F15.3)
      IF(PCODE(11)-2) 612,630,612
  612 IDTS=IDS(KSTATE(1))
      CALL UVWTG2(S,DATPRM(1,IDTS),PHI,FLAM,H)
      PHI=PHI*RPD
      FLAM=FLAM*RPD
      WRITE(6,6023) PHI,FLAM,H
 6023 FORMAT('  GEOD. COORD. OF SATELLITE',2F14.6,F14.1)
  630 CONTINUE
      WRITE(6,6012) GQI
 6012 FORMAT(10X,'GQI=',F10.5)
      IF(ECODE.GT.1) GO TO 290
      IF(NSUSED.EQ.0) GO TO 290
      RMSMC=DSQRT(EVSUM/DFLOAT(NSUSED))
      WRITE(6,6011) RMSMC
 6011 FORMAT(1H+,27X,'RMS MISCLOSURE IN METERS=',F10.1)
      GO TO 300
  290 WRITE(6,6015) ECODE
 6015 FORMAT(1H+,27X,'ENTIRE EVENT DELETED, KODE=',I4)
      GO TO 600
C
C     SET UP OBSERVATION EQUATIONS FOR THIS EVENT AND COMPUTE CONTRIBUTOONS
C        TO THE NORMAL EQUATIONS
  300 CONTINUE
      IF(ECODE.GT.1) GO TO 600
      KEVENT=KEVENT+1
      DO 310 I=1,3
      DDK(I)=0.0
      DO 310 J=1,3
      DDN(I,J)=0.0
  310 CONTINUE
C
      JS=0
```

```
      DO 390 IS=1,NSTE
      IF(IPASS(IS).GT.1) GO TO 390
      JS=JS+1
C        JS IS THE COUNTER FOR NON-DELETED STATIONS IN THE EVENT
      RSQCSD=SDC(1,IS)**2+SDC(2,IS)**2
      RSQ=RSQCSD+SDC(3,IS)**2
      RCD=DSQRT(RSQCSD)
      ASC=DATAN2(SDC(2,IS),SDC(1,IS))+GAST
      DSC=DATAN(SDC(3,IS)/RCD)
      DL(1)=ALFS(IS)-ASC
      IF(DL(1).GT.PI) DL(1)=DL(1)-PI2
      IF(DL(1).LT.(-PI)) DL(1)=DL(1)+PI2
      DL(2)=DEC(IS)-DSC
C
COMPUTE WEIGHTS
      VARRA(IS)=(VARRA(IS)/SPR)**2
      VARRA(IS)=VARRA(IS)*RSQ/RSQCSD
      VARDEC(IS)=(VARDEC(IS)/SPR)**2
      COVRAD(IS)=COVRAD(IS)/SPR**2
      DET=VARRA(IS)*VARDEC(IS)-COVRAD(IS)**2
      W(1,1)=VARDEC(IS)/DET
      W(2,2)=VARRA(IS)/DET
      W(1,2)=-COVRAD(IS)/DET
      W(2,1)=W(1,2)
C
COMPUTE OBSERVATION EQUATIONS
      A(1,1)=-SDC(2,IS)/RSQCSD
      A(1,2)=+SDC(1,IS)/RSQCSD
      A(1,3)=0.0
      FACTOR=SDC(3,IS)/(RSQ*RCD)
      A(2,1)=-SDC(1,IS)*FACTOR
      A(2,2)=-SDC(2,IS)*FACTOR
      RANGE=DSQRT(RSQ)
      A(2,3)=RCD/RSQ
      CALL DGMPRD(A,PM,AP,2,3,3)
 6017 FORMAT(1X,7D17.9)
C
      KSTA=KSTATE(IS)
C   ELIMINATE  DELETED STATIONS FROM THE LIST OF STATIONS INVOLVED IN
C       THE EVENT.
      KSTATE(JS)=KSTATE(IS)
C
COMPUTE VPV OF MISCLOSURES
      VPVTO=0.0
      DO 315 II=1,2
      DO 315 JJ=1,2
  315 VPVTO=DL(II)*W(II,JJ)*DL(JJ)+VPVTO
      NOBSTA(KSTA)=NOBSTA(KSTA)+2
      VPVSTA(KSTA)=VPVSTA(KSTA)+VPVTO
COMPUTE CONTRIBUTION TO NORMALS
      DO 330 I=1,3
      DO 325 J=1,3
      TERM=0.0
      DO 320 II=1,2
      DO 320 JJ=1,2
  320 TERM=TERM+AP(II,I)*W(II,JJ)*AP(JJ,J)
      BN(I,J,JS)=-TERM
      DN(I,J,KSTA)=DN(I,J,KSTA)+TERM
```

```
      DDN(I,J)=DDN(I,J)+TERM
  325 CONTINUE
C
      TERM=0.0
      DO 328 II=1,2
      DO 328 JJ=1,2
  328 TERM=TERM+AP(II,I)*W(II,JJ)*DL(JJ)
      DK(I,KSTA)=DK(I,KSTA)-TERM
      DDK(I)=DDK(I)+TERM
  330 CONTINUE
C
  390 CONTINUE
C INVERT DDN
      DET=1.0
      CALL DMINV(DDN,3,DET,L1,L2)
      CALL DGMPRD(DDK,DDN,TA,1,3,3)
      CALL DGMPRD(TA,DDK,TB,1,3,1)
      WPWSP=WPWSP+TB
      NSUSED=JS
      WRITE(2) NSUSED,DDN,DDK,(((BN(I,J,JS),I=1,3),J=1,3),KSTATE(JS),
     1JS=1,NSUSED),CONTIN
  600 CONTINUE
C
C     TEST FOR END OF INPUT
      IF(CONTIN.EQ.ENDSIG) GO TO 700
      GO TO 210
C
C
C
  700 CONTINUE
C
CHECK TO SEE IF END SIGNAL HAS BEEN WRITTEN ON DATA SET FT02
      IF(ECODE.EQ.1) GO TO 710
      BACKSPACE 2
C   READ AND REWRITE LAST RECORD FROM LAST GOOD EVENT
      READ (2) NSUSED,DDN,DDK,(((BN(I,J,JS),I=1,3),J=1,3),KSTATE(JS),
     1JS=1,NSUSED)
      BACKSPACE 2
      WRITE(2) NSUSED,DDN,DDK,(((BN(I,J,JS),I=1,3),J=1,3),KSTATE(JS),
     1JS=1,NSUSED),CONTIN
  710 CONTINUE
      WRITE(2)       ((( DN(I,J,KSTA),I=1,3),DK(J,KSTA),J=1,3),
     XKSTA=1,NSTA)
C     WRITE(6,6018)(KORDER(KSTA),((DN(I,J,KSTA),J=1,3),I=1,3),
C    1KSTA=1,NSTA)
 6018 FORMAT((I5/3(3D18.7/)))
      WPW=0.0
      NOBS=0
      WRITE(6,6019)
 6019 FORMAT(1H1,8(/),10X,'ANALYSIS OF MISCLOSURES BY STATION'//
     1T10,'STATION',T20,'NUMBER OF OBSERVATIONS',T50,'RMS MISCLOSURE')
      DO 750 KSTA=1,NSTA
      NOBS=NOBS+NOBSTA(KSTA)
      WPW=WPW+VPVSTA(KSTA)
      RMSMC=0.0
      IF(NOBSTA(KSTA).GT.0) RMSMC=DSQRT(VPVSTA(KSTA)/DFLOAT(NOBSTA(KSTA)
     1))
      WRITE(6,6020) KORDER(KSTA),NOBSTA(KSTA),RMSMC
```

```
6020 FORMAT(T10,I7,T35,I7,T50,F14.2)
 750 CONTINUE
     IDEGF=NOBS-3*KEVENT
     RMSMC=DSQRT(WPW/DFLOAT(IDEGF))
     WRITE(6,6021) NOBS,KEVENT,IDEGF,WPW,RMSMC
6021 FORMAT(////10X,'TOTAL NUMBER OF GOOD OBSERVATIONS',T60,I8//
    110X,'TOTAL NUMBER OF GOOD EVENTS',T60,I8,//
    210X,'CORRESPONDING DEGREES OF FREEDOM',T60,I8//
    310X,'TOTAL SUM OF SQUARES OF MISCLOSURES',T60,F11.2//
    410X,'CORRESPONDING STANDARD DEVIATION OF UNIT WEIGHT',T60,F11.2)
     WPW=WPW-WPWSP
     WRITE(6,6022) WPW
6022 FORMAT(1H0,9X,'WPW INCLUDING CONTRIBUTION FROM SATELLITE POSITION'
    1/15X,'(I.E., VPV+UX)',T60,F11.2)
     RETURN
     END
```

```
      SUBROUTINE EXPAND (XPO,YPO,ZPO,CNM,SNM,LCT,ICT,UMT,VMT,CTB,CTT
     1,ERD,XMU,ALF,OMG,ECC,NTE,KTR,KDR,NHT,CDC,CTW,KEY,DMT,KRG,CMC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LT(5),KT(5)
      DIMENSION XPO(1),YPO(1),ZPO(1),CNM(1),SNM(1)
     1,  LCT(1),ICT(1),UMT(1),VMT(1),CTB(1),CTT(1)
      DIMENSION DMT(1)
      DIMENSION CMC(3)
COMMON INTERNAL WORKING ARRAYS
      DIMENSION
     1 XTL(8),CLB(8),CLT(8),RPT(8),RMO(8),RZR(8),VRB(8),
     2 YTL(8),SLB(8),SLT(8),RMT(8),RMR(8)
      COMMON   /ORBCOM/
     1CSQ(8),SSQ(8),SCS(8),BXB(8),BYB(8),BZB(8),XVR(8),YVR(8),ZVR(8),
     2 QAV(8),QBV(8),QCV(8),AMT(72),FEE(72),CGB(8),SGB(8)
      EQUIVALENCE
     1 (CLB(1),CSQ(1)),(CLT(1),XVR(1)),(RPT(1),BXB(1)),(RZR(1),QAV(1))
     2,(SLB(1),SSQ(1)),(SLT(1),YVR(1)),(RMT(1),BYB(1)),(RMR(1),QBV(1))
     3,(XTL(1),SCS(1)),(YTL(1),ZVR(1)),(RMO(1),BZB(1)),(VRB(1),QCV(1))
COMMON DATA BLOCK END
      EQUIVALENCE (LA,LT(1)),(LB,LT(2)),(LC,LT(3)),(K,LT(4)),(L,LT(5))
      EQUIVALENCE (KA,KT(1)),(KB,KT(2)),(KC,KT(3)),(M,KT(4)),(N,KT(5))
C
      CGB(1)=DCOS(ALF)
      SGB(1)=DSIN(ALF)
      CLB(1)=XPO(1)*CGB(1)+YPO(1)*SGB(1)
      SLB(1)=YPO(1)*CGB(1)-XPO(1)*SGB(1)
      RPT(1)=XPO(1)*XPO(1)+YPO(1)*YPO(1)+ZPO(1)*ZPO(1)
      RMT(1)=1.0/RPT(1)
      RMO(1)=DSQRT(RMT(1))
      CLT(1)=ERD*RMT(1)*CLB(1)
      SLT(1)=ERD*RMT(1)*SLB(1)
      RZR(1)=ERD*RMT(1)*ZPO(1)
      RMR(1)=ERD*ERD*RMT(1)
      DRX=0.0
      DRY=0.0
      DRZ=0.0
      NG = 0
      ITR=3
      KA=1
      IF (KEY-2) 10,10,490
COMPUTE CTB,CTT ARRAYS TO MATCH TABLE ALLOCATION
   10 CONTINUE
COMPUTATION OF CTB AND CTT ARRAYS MOVED TO DATA STATEMENT IN ORBIT
      GO TO (60,150),KEY
COMPUTE ICT TABLE ENTRIES AND TOTAL ARRAY LENGTH
   60 CONTINUE
      KEY=3
      NET=1
      KTR=10
      LNTH=1
      LA=LCT(2)
      LB=LCT(1)
      LC=LA
      NA=NTE+1
      DO 130 I=1,NA
      KA=LA-1
      KB=LB
```

```
      KC=LC+1
      IF (KB-KA) 70,80,80
   70 KB=KA
   80 IF (LC) 110,110,90
   90 IF (KB-KC) 100,110,110
  100 KB=KC
  110 LNTH=LNTH+KB+1
      ICT(I)=LNTH
      LA=LB
      LB=LC
      LC=LCT(I+2)
      IF (I-NTE) 130,120,120
  120 LC=0
  130 CONTINUE
      LNTH=LNTH-1
      KA=1
      GO TO 490
  150 CONTINUE
      KEY=3
      NET=2
      LNTH=1
      DO 160 I=1,3
  160 LT(I+2)=LCT(I)
      IF (NTE-2)170,180,190
  170 K=0
  180 L=0
  190 LB=K
      LA=L
      NA=NTE+2
      DO 280 J=1,NA
      DO 210 I=1,5
      KT(I)=0
      IF (LT(I))210,210,200
  200 KT(I)=LT(I)+I-2
  210 CONTINUE
      KK=0
      DO 230 I=1,5
      IF (KK-KT(I))220,230,230
  220 KK=KT(I)
  230 CONTINUE
      LNTH=LNTH+KK+1
      ICT(J)=LNTH
      DO 260 I=1,4
  260 LT(I)=LT(I+1)
      L=LCT(J+3)
      N=J-NTE+2
      IF (N)280,270,270
  270 L=0
  280 CONTINUE
      LNTH=LNTH-1
      KA=1
      GO TO 490
C  BEGIN THE ITERATIVE INTEGRATION FOR THE SOLUTION.
C    300 SERIES - COMPUTE X,Y,Z COEFFICIENTS
  300 KA=1+(ITR-3)*LNTH
      KB=1
      TMA=0.0
      TMB=0.0
```

```
      TMC=0.0
      TMD = 0.5
      KG = 0
      NA = ICT(1)-2
      NB = 1
      NC = NA+2
      DO 380 I=1,NTE
      KC = LCT(I)
      IF (KC)380,380,310
  310 IF (I-2)330,315,320
  315 NA = 1
      GO TO 325
  320 NA = ICT(I-2)
  325 NB = ICT(I-1)
      NC = ICT(I)
  330 LA = KA+NA+1
      LB = KA+NB
      LC = KA+NC-1
      DO 375 J=1,KC
      TME = CTB(KB)
      TMF = TME
      IF (KG)345,345,350
  345 TMF = TMD*TME
      TME = -TMF
      TMI = J
      TMJ = J+2
      TMD = TMD*TMI/TMJ
  350 TME = TME*UMT(LA)
      TMF = TMF*VMT(LA)
      TMG = UMT(LC)
      TMH = VMT(LC)
      TMI = CNM(KB)
      TMJ = SNM(KB)
      TMK = TME-TMG
      TML = TMF+TMH
      TMM = TMF-TMH
      TMN = TME+TMG
      TMO = UMT(LB)
      TMP = VMT(LB)
      IF (KRG)370,370,365
  365 DMT(NG+1) = TMK
      DMT(NG+2) = TML
      DMT(NG+3) = TMO
      DMT(NG+4) = TMM
      DMT(NG+5) = TMN
      DMT(NG+6) = TMP
      NG = NG+6
  370 TMA = TMA+TMI*TMK+TMJ*TMM
      TMB = TMB-TMI*TML+TMJ*TMN
      TMC = TMC+CTT(KB)*(TMI*TMO+TMJ*TMP)
      LA = LA+1
      LB = LB+1
      LC = LC+1
      KB = KB+1
  375 CONTINUE
  380 KG = 1
      TMD = 0.5/ERD
      KK=ITR-2
```

```
      XTL(KK)=TMA*TMD
      YTL(KK)=TMB*TMD
      TMC=TMC*TMD
      TMD=KK*(KK+1)
      KA=KK
      TMA=DRX
      TMB=DRY
      TMC=TMC+DRZ
      DO 390 I=1,KK
      TME=CGB(I)
      TMF=SGB(I)
      TMG=XTL(KA)
      TMH=YTL(KA)
      TMA=TMA+TME*TMG-TMF*TMH
      TMB=TMB+TMF*TMG+TME*TMH
  390 KA=KA-1
      TMA=TMA-OMG*OMG*(CMC(1)*CGB(KK)-CMC(2)*SGB(KK))
      TMB=TMB-OMG*OMG*(CMC(1)*SGB(KK)+CMC(2)*CGB(KK))
      XPO(ITR)=TMA/TMD
      YPO(ITR)=TMB/TMD
      ZPO(ITR)=TMC/TMD
      ITR=ITR+1
      IF (KTR-ITR)400,410,410
  400 RETURN
C     400 SERIES - COMPUTE EXTENSION DERIVATIVES
  410 TMD=KK
      KA=KK+1
      CGB(KA)=-OMG*SGB(KK)/TMD
      SGB(KA)= OMG*CGB(KK)/TMD
      KK=KA
      KB=KK
      TMA=0.0
      TMB=0.0
      TMC=0.0
      DO 420 I=1,KK
      TME=XPO(KB)
      TMF=YPO(KB)
      TMG=CGB(I)
      TMH=SGB(I)
      TMA=TMA+TME*TMG+TMF*TMH
      TMB=TMB+TMF*TMG-TME*TMH
      TMC=TMC+TME*XPO(I)+TMF*YPO(I)+ZPO(KB)*ZPO(I)
  420 KB=KB-1
      CLB(KA)=TMA
      SLB(KA)=TMB
      RPT(KA)=TMC
      KB=KK-1
      RMT(KA)=0.0
      TMA=0.0
      DO 430 I=2,KK
      TMA=TMA-RMT(KB)*RPT(I)
  430 KB=KB-1
      RMT(KA)=RMT(1)*TMA
      KB=KK-1
      TMA=RMT(KA)
      KC=KB
      IF (KC-1)460,460,440
  440 DO 450 I=2,KC
```

```
      TMA=TMA-RMO(KB)*RMO(I)
  450 KB=KB-1
  460 RMO(KA)=TMA/(2.0*RMO(1))
      KB=KK
      TMA=0.0
      TMB=0.0
      TMC=0.0
      DO 470 I=1,KK
      TMI=RMT(KB)
      TMA=TMA+TMI*CLB(I)
      TMB=TMB+TMI*SLB(I)
      TMC=TMC+TMI*ZPO(I)
  470 KB=KB-1
      CLT(KA)=TMA*ERD
      SLT(KA)=TMB*ERD
      RMR(KA)=RMT(KA)*ERD*ERD
      RZR(KA)=TMC*ERD
C   500 SERIES - COMPUTE U,V ARRAY EXTENSION COEFFICIENTS
  490 CONTINUE
      NA=LNTH*(ITR-3)+1
      UMT(NA)=XMU*RMO(KA)
      VMT(NA)=0.0
      LA=1
      LB=1
      KB=2
      NB=NTE+NET
      DO 580 I=1,NB
      KC=ICT(I)-1
      M=I-1
      N=M
      IF (KC-KB)550,500,500
  500 KK=2
      DO 540 J=KB,KC
      LC=LB
      NC=KA
      TMA=0.0
      TMB=0.0
      TMC=0.0
      TMD=0.0
      TME=N+N+1
      TMF=N+M
      TMG=N-M+1
      DO 530 K=1,KA
      GO TO (510,520),KK
  510 TMI=RMR(NC)
      TMC=TMC+TMI*UMT(LC-1)
      TMD=TMD+TMI*VMT(LC-1)
  520 TMJ=RZR(NC)
      TMA=TMA+TMJ*UMT(LC)
      TMB=TMB+TMJ*VMT(LC)
      NC=NC-1
  530 LC=LC+LNTH
      NA=NA+1
      UMT(NA)=(TME*TMA-TMF*TMC)/TMG
      VMT(NA)=(TME*TMB-TMF*TMD)/TMG
      LB=LB+1
      N=N+1
  540 KK=1
```

```
  550 IF (I-NB)560,580,580
  560 NA=NA+1
      TMC=2*M+1
      NC=KA
      TMA=0.0
      TMB=0.0
      DO 570 K=1,KA
      TMG=CLT(NC)
      TMH=SLT(NC)
      TMI=UMT(LA)
      TMJ=VMT(LA)
      TMA=TMA+TMG*TMI-TMH*TMJ
      TMB=TMB+TMG*TMJ+TMH*TMI
      LA=LA+LNTH
  570 NC=NC-1
      UMT(NA)=TMA*TMC
      VMT(NA)=TMB*TMC
      KB=KC+2
      LB=LB+1
      LA=LB
  580 CONTINUE
      IF (KDR) 300,300,600
CODING FOR DRAG COMPUTATIONS ARE OMITED
  600 GO TO 300
      END
```

```
      SUBROUTINEVARIEQ(XPO,YPO,ZPO,CNM,SNM,LCT,ICT,UMT,VMT,UVM,ERD,ALF,
     *OMG,CDC,CTW,NTE,KTR,KDR)
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONTMT(9)
      DIMENSIONXPO(1),YPO(1),ZPO(1),CNM(1),SNM(1),LCT(1),ICT(1),UMT(1),
     *VMT(1),UVM(1)
COMMON INTERNAL WORKING ARRAYS
      DIMENSION XTL( 8),CLB( 8),CLT( 8),RPT( 8),RMO( 8),RZR( 8),VRB( 8),
     *  YTL( 8),SLB( 8),SLT( 8),RMT( 8),RMR( 8)
      COMMON/ORBCOM/CSQ( 8),SSQ( 8),SCS( 8),BXB( 8),BYB( 8),BZB( 8),
     1 XVR( 8),YVR( 8),ZVR( 8),QAV( 8),QBV( 8),QCV( 8),
     2  AMT( 72),FEE( 72),CGB( 8),SGB( 8)
      EQUIVALENCE(CLB(1),CSQ(1)),(CLT(1),XVR(1)),(RPT(1),BXB(1)),(RZR(1
     *),QAV(1)),(SLB(1),SSQ(1)),(SLT(1),YVR(1)),(RMT(1),BYB(1)),(RMR(1),
     *QBV(1)),(XTL(1),SCS(1)),(YTL(1),ZVR(1)),(RMO(1),BZB(1)),(VRB(1),QC
     *V(1))
COMMON DATA BLOCK END
      EQUIVALENCE(TMT(1),TMA),(TMT(2),TMB),(TMT(3),TMC)
      EQUIVALENCE(TMT(4),TMD),(TMT(5),TME),(TMT(6),TMF)
      EQUIVALENCE(TMT(7),TMG),(TMT(8),TMH),(TMT(9),TMI)
C
C INITIALIZE ALL OF THE EXTERNAL ARRAYS.
C
      CALLCLEAR(TMT,9)
      IF(KTR-2)90,90,20
   20 KA=KTR-2
      DO40KC=1,KA
      KB=KC
      DO30I=1,KC
      AU=CGB(KB)
      BU=SGB(I)
      TMA=TMA+CGB(I)*AU
      TMB=TMB+BU*SGB(KB)
      TMC=TMC+BU*AU
   30 KB=KB-1
      CSQ(KC)=TMA
      SSQ(KC)=TMB
      SCS(KC)=TMC
      TMA=0.0
      TMB=0.0
      TMC=0.0
   40 CONTINUE
      RDS=ERD*ERD
      LNC=ICT(NTE+2)-1
      ITR=2
      L=9*KTR-18
      CALLCLEAR(AMT,L)
      KDG=KDR-2
C   DRAG  COMPUTATIONS DELETED. INITIALIZATION OF HNV AND VRN DELETED.
C
C  TEST FOR PROCESS COMPLETE - BEGIN THE ITERATIVE LOOP
C
   80 ITR=ITR+1
      IF(KTR-ITR)90,100,100
   90 RETURN
  100 KA=ITR-3
      I=1+KA*LNC
      LA=I+ICT(2)-1
```

```
      LB=I+ICT(1)
      LC=I+2
      LD=LB
      LE=LA
      M=0
      KB=1
      TMQ=0.0
      TMR=1.0/24.0
      TMS=1.0/6.0
      ICT2=ICT(2)
      DO270I=1,NTE
      N=M
      KC=LCT(I)
      IF(KC)210,210,120
  120 CONTINUE
      DO200J=1,KC
      L=N-M+2
      CZZ=L*(L-1)
      CPZ=L*(L+1)
      CPM=(L+1)*(L+2)
      AU=UMT(LA)
      BU=UMT(LB)
      CU=UMT(LC)
      DU=UMT(LD)
      EU=UMT(LE)
      AV=VMT(LA)
      BV=VMT(LB)
      CV=VMT(LC)
      DV=VMT(LD)
      EV=VMT(LE)
      CC=CNM(KB)
      SC=SNM(KB)
      TMI=TMI+CZZ*(CC*CU+SC*CV)
      IF(M-1)140,160,180
  140 AU=AU*TMR
      BU=-BU*TMS
      AV=-AV*TMR
      BV=BV*TMS
      TMQ=TMQ+1.0
      TMR=TMR*TMQ/(TMQ+4.0)
      TMS=TMS*(TMQ+1.0)/(TMQ+3.0)
      GOTO180
  160 AU=-AU*TMS
      AV=AV*TMS
      TMQ=TMQ+1.0
      TMS=TMS*TMQ/(TMQ+2.0)
  180 T=L-1
      TMA=TMA+CC*(EU+CZZ*(CPM*AU-CU-CU))+SC*(EV+CZZ*(CPM*AV-CV-CV))
      TMB=TMB+CC*(EV-CZZ*CPM*AV)+SC*(CZZ*CPM*AU-EU)
      TMC=TMC-T*(CC*(CPZ*BU-DU)+SC*(CPZ*BV-DV))
      TMF=TMF+T*(CC*(CPZ*BV+DV)-SC*(CPZ*BU+DU))
      KB=KB+1
      LA=LA+1
      LB=LB+1
      LC=LC+1
      LD=LD+1
      LE=LE+1
  200 N=N+1
```

```
  210 L=KA*LNC
      IF(M-1)220,230,240
  220 TMQ=2.0
      TMS=1.0/12.0
      LA=L+ICT2+2
      LB=L+4
      GOTO260
  230 LA=L+5
      GOTO250
  240 LA=L+4+ICT(I-1)
  250 LB=L+3+ICT(I)
  260 LC=L+2+ICT(I+1)
      LD=L+1+ICT(I+2)
      LE=L+ICT(I+3)
  270 M=M+1
      TMR=2.0*RDS
      TMC=TMC/TMR
      TMF=TMF/TMR
      TMR=2.0*TMR
      TMA=TMA/TMR
      TMB=TMB/TMR
      TMD=TMB
      TMG=TMC
      TMH=TMF
      TMI=TMI/RDS
      TME=-TMA-TMI
      L=9*KA+1
      DO280I=1,9
      FEE(L)=TMT(I)
  280 L=L+1
      CALLCLEAR(TMT,9)
C
C  EVALUATE THE K-TH TERM OF THE MATRIX A(3,3)
C
C   DRAG COMPUTATIONS DELETED.
  500 L=0
      KB=KA+1
      N=KB
      DO510I=1,KB
      AU=FEE(L+1)
      BU=FEE(L+2)
      CU=FEE(L+3)
      DU=FEE(L+5)
      EU=FEE(L+6)
      AV=CSQ(N)
      BV=SSQ(N)
      CV=SCS(N)
      DV=CGB(N)
      EV=SGB(N)
      TMA=TMA+AU*AV-2.0*BU*CV+DU*BV
      TMB=TMB+CV*(AU-DU)+BU*(AV-BV)
      TMC=TMC+CU*DV-EU*EV
      TMF=TMF+CU*EV+EU*DV
      N=N-1
  510 L=L+9
      L=9*KA+9
      TMI=FEE(L)
      TMD=TMB
```

```
      TME=-TMA-TMI
      TMG=TMC
      TMH=TMF
      L=L-8
      DO520I=1,9
      AMT(L)=AMT(L)+TMT(I)
  520 L=L+1
      CALLCLEAR(TMT,9)
C
C  EVALUATE U AND V MATRICES FOR (K+2)TH TERM IN SERIES
C
      KB=KA+1
      TMQ=KB*(KB+1)
      TMQ=1.0/TMQ
      TMR=1.0
      KC=18*KA-2
      DO600LA=1,2
      LB=KC+39
      LC=0
      DO580LD=1,KB
      LE=1
      DO570J=1,3
      I=3*J+KC
      AU=UVM(I)
      BU=UVM(I+1)
      CU=UVM(I+2)
C   DRAG COMPUTATIONS DELETED.
      DO570M=1,3
      I=LC+M
      TMS=AMT(I)*AU+AMT(I+3)*BU+AMT(I+6)*CU
C   DRAG COMPUTATIONS DELETED.
  560 TMT(LE)=TMT(LE)+TMS
  570 LE=LE+1
      KC=KC-18
      TMR=TMR+1.0
  580 LC=LC+9
      DO590I=1,9
      UVM(LB)=TMQ*TMT(I)
  590 LB=LB+1
      CALLCLEAR(TMT,9)
      TMR=1.0
  600 KC=18*KA+7
      GOTO80
      END
```

```
      SUBROUTINE ORBIN
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION EMODEL(10,2)
      DIMENSION XIN(6),ORBEL(6),RO(3),VO(3),ORBNAM(6),ORBUNK(16),S(3,3)
      EQUIVALENCE (ORBEL(1),ORBA),(ORBEL(2),ORBECC),(ORBEL(3),ORBINC),
     1(ORBEL(4),RANODE),(ORBEL(5),ARGPGE),(ORBEL(6),ORBM)
      EQUIVALENCE (XIN(1),RO(1)),(XIN(4),VO(1)),(THEDOT,OMGI)
      INTEGER*2 LFLG(40),KSTATO(15),MODEL(10,2)
      INTEGER*2 GUIDE(10,2),IDAY,IYR
      INTEGER*4 ENDSIG/1HE/,CONTIN,BLANK/1H /,ALFA/1HA/,ALFR/1HR/,ZCODE
      DATA MAXSTO/15/,MAXSTA/40/,MAXEMU/10/
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      COMMON/OBSD/OBSD(150),OVOBSD
      COMMON/ERDCON/ERDI,XMUI,OMGI,XCM,YCM,ZCM
      COMMON/NSTA/NSTA
      COMMON/STAORD/KORDER(150)
      INTEGER STANAM,IDS*2
      COMMON/STALOC/STAUVW(3,150),DATDUM(6,15),STANAM(5,150),IDS(150)
      DIMENSION CM(3)
      EQUIVALENCE(CM(1),XCM)
C   READ EARTH CONSTANTS
      READ(5,1065) ERDI,XMUI,OMGI,OVOBSD
 1065 FORMAT(4D20.8)
C   READ THE COORDINATES OF THE CENTER OF MASS IN THE COORDINATE SYSTEM IN WHICH
C   THE STATION COORDINATES ARE GIVEN.
      READ(5,1052) RO
C   READ UNCERTAINTIES OF CENTER OF MASS
      READ(5,1052) VO
      WRITE(6,1063) RO,VO
 1063 FORMAT(/////'   COORDINATES OF THE CENTER OF MASS',3F20.3/
     1            '   UNCERTAINTIES                     ',3F20.3)
      WRITE(6,1064)      ERDI,XMUI,OMGI
 1064 FORMAT(1H1///'   EARTH CONSTANTS FOR ORBIT INTEGRATION'/
     1         T7,'SEMI-MAJOR AXIS',T30,'GM',T48,'ROTATION RATE'/
     2  F20.3,1PD20.10,0PD20.10)
      WRITE(3) ERDI,XMUI,OMGI,RO,VO
CHANGE THE DATUM ID NUMBER IF RESULTS WILL BE IN AN EARTH CENTERED DATUM
      IF(RO(1).EQ.0) GO TO 135
      IERD=ERDI
      I=0
  132 I=I+1
      IDAT=DATPRM(1,I)
      IF(IDAT.EQ.IERD) GO TO 133
      IF(I.LT.15) GO TO 132
      WRITE (6,1001)
 1001 FORMAT(//10X,'EARTH DIAMETER DOES NOT CORRESPOND TO ANY KNOWN DATU
     1M'//)
      GO TO 135
  133 DO 134 J=1,150
  134 IDS(J)=I
  135 CONTINUE
      DO 130 I=1,3
  130 CM(I)=RO(I)
      IF(PCODE(14).EQ.0) GO TO 140
      DO 139 I=1,3
  139 STAUVW(I,NSTA+1)=RO(I)
      KORDER(NSTA+1)=0
```

```
  140 CONTINUE
      SPR=206264.80625
      NOBS=0
      NUNK=0
      DO 160  I=7,16
 160  ORBUNK(I)=0
C               READ FIRST CARD
  165 READ(5,1051) NORB,ORBNAM,IOCODE,CONTIN
 1051 FORMAT(A4,6A8,I1,26X,A1)
C               TEST FOR END OF ALL PASSES
      IF(CONTIN.EQ.ENDSIG)  GO TO210
      WRITE(6,1062) NORB,ORBNAM
 1062 FORMAT(7(/),1X,A4,5X,6A8)
C
C         IOCODE=0  MEANS  RECTANGULAR ELEMENTS ARE GIVEN IN TRUE SIDEREAL SYSTEM
C         IOCODE=1  MEANS  RECTANGULAR ELEMENTS ARE GIVEN IN MODIFIED SIDEREAL
C           SYSTEM.
C         IOCODE=2  MEANS  RECTANGULAR ELEMENTS ARE GIVEN IN EARTH FIXED SYSTEM
C         IOCODE=3 MEANS KEPLERIAN ELEMENTS ARE GIVEN, REFERED TO TRUE EQUINOX.
C         IOCODE=4 MEANS KEPLERIAN ELEMENTS ARE GIVEN, REFERED TO TRUE EQUATOR
C          AND 1950 EQUINOX (I.E., THE SOA ORBITAL SYSTEM).
C
      DO 166 I=1,40
  166 LFLG(I)=0
      DO 167 I=1,10
      DO 167 J=1,2
      GUIDE(I,J)=00.0
  167 CONTINUE
      NUTORB=6
      NEMUNK=0
      IF(IOCODE.LT.3)GO TO 170
      READ(5,1052) ORBA,ORBECC,ORBINC
      READ(5,1052) RANODE,ARGPGE,ORBM
 1052 FORMAT(3D15.8)
      WRITE(6,1055) ORBEL
 1055 FORMAT(1X,'A=',D18.8,'ECC=',D18.8,'INC=',D18.8,/'RA OF NODE=',
     1D15.8,'ARG OF PERIGEE=',D15.8,'MEAN ANOMALY=',D15.8)
      CALL KEPTCE(ORBEL,XIN)
      IOCODE=IOCODE-3
      GO TO 171
  170 READ(5,1052) RO,VO
C     THERE ARE THREE WAYS THE EPOCH CAN APPEAR,AS  JULIAN DAYS IN COLUMNS
C     0 THRU 15,AS A DATE IN COLUMNS 16 THRU 35,OR IF A DATE OTHER THAN THAT
C     IS TO BE USUED THE DATE OF THE ORBITAL ELIMENTS GIVEN WILL BE IN
C     COLUMNS 16 THRU 35 WITH THE DESIRED EPOCH TIME IN COLUMNS 51 THRU 71
C     ZCODE COLUMNS 36 THRU41 MUST HAVE SOMETHING PUNCHED IN IT IN THE LAST CASE
  171 READ(5,1053) EPOCH,IDAY,MONTH,IYR,IH,MIN,ESEC,ZCODE,IHR,IMIN,SEC
 1053 FORMAT(D15.8,I5,2X,A3,3I5,D10.5,3X,A2,T51,2I5,D10.5)
      IF(EPOCH.EQ.0) GO TO 175
      ITEM=EPOCH
      TEM=ITEM
      TE=(EPOCH-TEM)*86400
      TEMP=TE/3600.0
      IH=TEMP
      TEMP=(TEMP-IH)*60.0
      MIN=TEMP
      ESEC=(TEMP-MIN)*60.0
      TO=IHR*3600+IMIN*60+SEC
```

```
      TIME=EPOCH
      GO TO 180
  175 TEM=MJD(IDAY,MONTH,IYR)
      TE=IH*3600+MIN*60+ESEC
      TO=IHR*3600+IMIN*60+SEC
      EPOCH=TEM+TE/86400.0
      TIME=TEM+TO/86400.0
      IF(ZCODE.EQ.BLANK) TIME=EPOCH
  180 CONTINUE
      WRITE(6,1054) RO,VO,EPOCH
      IF(IOCODE.NE.1) GO TO 185
      CALL PRENUT(TIME,PAN)
      CALL ROT3(-PAN,RO)
      CALL ROT3(-PAN,VO)
  185 GASTO=GSTD(TIME)
      CALL PRENUT(TIME,PAN)
      GASTO=GASTO+PAN
 1054 FORMAT(/1X,'X=',D18.8,1X,'Y=',D18.8,1X,'Z=',D18.8,/1X,'XDOT=',
     1D15.8,1X,'YDOT=',D15.8,1X,'XDOT=',D15.8,/1X,'EPOCH=',D15.8,10X,
     2I5,2X,A3,I5,/22X,'GAST=',D15.8//)
      IF(IOCODE.NE.2) GO TO 190
      RO1=RO(1)
      RO2=RO(2)
      CALL ROT3(-GASTO,RO)
      VO(1)=VO(1)-THEDOT*RO1
      VO(2)=VO(2)+THEDOT*RO2
      CALL ROT3(-GASTO,VO)
  190 IF(TO-TE) 195,200,196
  195 IF(ZCODE.EQ.BLANK) GO TO 200
  196 IF(TE.NE.0) GO TO 198
      EPOCH=TIME
      GO TO 200
  198 CALL ORBIT(0,TO,TE,GASTO,XIN)
  200 CALL POLE(EPOCH,XPM,YPM)
      XPM=XPM/SPR
      YPM=YPM/SPR
      XP=XPM
      YP=YPM
      GAST=GSTD(EPOCH)
      CALL PRENUT(EPOCH,PAN)
      GAST=GAST+PAN
      WRITE(6,1154)
 1154 FORMAT(/1X,'VALUES STORED ON UNIT 3')
      WRITE(6,1054)RO,VO,EPOCH,IDAY,MONTH,IYR,GAST
  210 WRITE(3)IORB,NORB,ORBNAM,EPOCH,IDAY,MONTH,IYR,GAST,RO,VO,TE,IH,
     1MIN,ESEC,XPM,YPM,CONTIN
      IF(CONTIN.EQ.ENDSIG) RETURN
      NOBSTO=0
      WRITE(6,6005)
 6005 FORMAT(' STATION   DATE',T25,'TIME(UT)',3X,'OBSERVED  RANGE',2X,
     1   'UNCERTAINTY')
C                READ OBSERVATION CARD
  220 READ(5,1061)ID,IYR,MONTH,IDAY,IH,IMIN,SEC,SEC1,RSO,RSO1,VARRA,
     1CONTIN
 1061 FORMAT(14X,I4,5I2,F2.0,F4.0,F16.0,F3.0,11X,F6.3,9X,A1)
C                TEST FOR END OF PASS
      IF(CONTIN.EQ.ENDSIG) GO TO 240
      KSTA=KSTAID(ID)
```

```
      IF(CONTIN.EQ.ALFA) GO TO 290
      IF(CONTIN.EQ.ALFR) GO TO 300
      NOBSTO=NOBSTO+1
      NOBS=NOBS+1
      IF(PCODE(12).EQ.1) VARRA=OBSD(KSTA)
      IF(PCODE(12).EQ.2) VARRA=OVOBSD
      RSO=RSO+RSO1/1000.0
      IF(SEC1.LT.1) GO TO 230
      SEC=SEC+SEC1/10000.0
      GO TO 231
  230 SEC=SEC+SEC1
  231 CONTINUE
      LFLG(KSTA)=1
      TEMO=MJD(IDAY,MONTH,IYR)
      TOBS=IH*3600+IMIN*60+SEC+(TEMO-TEM)*86400.0
      TO=TOBS-TE
      IF(DABS(TO).GT.10000) GO TO 260
      GASTO=GAST+TO*THEDOT
      SINST=DSIN(GASTO)
      COSST=DCOS(GASTO)
C    FILL IN "S" ARRAY ACCORDING TO SAO SPECIAL REPORT 123,"PRECISE ASPECTS
C       OF TERRESTRIAL AND CELESTIAL REFERENCE FRAMES", PAGE 8
      S(1,1)=COSST
      S(2,1)=-SINST
      S(3,1)=-XP*COSST-YP*SINST
      S(1,2)=SINST
      S(2,2)=COSST
      S(3,2)=-XP*SINST+YP*COSST
      S(1,3)=XP
      S(2,3)=-YP
      S(3,3)=1.0
      WRITE(6,1056) ID,IDAY,MONTH,IYR,IH,IMIN,SEC,RSO,VARRA
 1056 FORMAT(I7,I4,1X,A3,I3,I5,I3,F8.4,F13.2,2F15.2)
  240 WRITE(3) TOBS,TO,KSTA,GASTO,SINST,COSST,RSO,VARRA,ID,IDAY,MONTH,
     1    IYR,IH,IMIN,SEC,S,CONTIN
      IF(CONTIN.EQ.ENDSIG) GO TO 270
      GO TO 220
  260 WRITE(6,1057) TOBS,TE
 1057 FORMAT(1X,'TIME OF OBSERVATION=',F20.10,/1X,'TIME OF EPOCH=',
     1F20.10)
      STOP
  270 NSTATO=0
      DO 280 KSTA=1,NSTA
      IF(LFLG(KSTA).EQ.0) GO TO 280
      NSTATO=NSTATO+1
      IF(NSTATO.LE.MAXSTO) GO TO 277
      WRITE(6,1058) MAXSTO
 1058 FORMAT(1X,'MORE THAN',I2,' STATIONS OBSERVED')
      STOP
  277 KSTATO(NSTATO)=KSTA
      LFLG(KSTA)=NSTATO
  280 CONTINUE
      IF(PCODE(14).EQ.0) GO TO 2800
      NSTATO=NSTATO+1
      KSTATO(NSTATO)=NSTA+1
 2800 CONTINUE
      DO 281 I=1,6
  281 ORBUNK(I)=XIN(I)
```

```
      IF(NEMUNK.EQ.0) GO TO 285
      DO 283 I=1,NEMUNK
C             GET THE ORDER # OF THE STATION TO WHICH THIS ERROR MODEL
C     TERM PERTAINS
      KSTA=MODEL(I,2)
      ISTA=LFLG(KSTA)
      IF(ISTA.NE.0) GO TO 282
      WRITE(6,1059) KORDER(KSTA)
 1059 FORMAT(1X,'ERROR MODEL UNKNOWN FOR STATION ',I3,' IS MEANINGLESS
     1SINCE STATION DOES NOT OBSERVE THIS PASS.')
  282 MODCOD=MODEL(I,1)
      GUIDE(ISTA,MODCOD)=I+6
  283 CONTINUE
  285 WRITE(4) ORBUNK,NSTATO,LFLG,KSTATO,NEMUNK,EMODEL,MODEL,GUIDE
     1   , NUTORB
      GO TO 165
  290 MODCOD=1
      GO TO 310
  300 MODCOD=2
      GO TO 310
  310 NEMUNK=NEMUNK+1
      IF(NEMUNK.LE.MAXEMU) GO TO 311
      WRITE(6,1060) MAXEMU
 1060 FORMAT(1X,'ERROR-NUMBER OF ERROR MODEL UNKNOWNS EXCEEDS ',I5)
      STOP
  311 NUTORB=NUTORB+1
      MODEL(NEMUNK,1)=MODCOD
      MODEL(NEMUNK,2)=KSTA
      EMODEL(NEMUNK,1)=RSO
      EMODEL(NEMUNK,2)=VARRA
      ORBUNK(NUTORB)=RSO
      GO TO 220
      END
```

```
      SUBROUTINE FOAN
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 IDAY,IYR
      INTEGER*4 CONTIN,ENDSIG/1HE/
      COMMON/NSTA/NSTA
      INTEGER STANAM,IDS*2
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/STAORD/KORDER(150)
      COMMON/WPW/WPW,XPU,IDEGF,NFSTA
      DIMENSION NOBSTA(40)
      REAL*4 VPVSTA(40)
      DIMENSION RO(3),VO(3),XT(6),UVWT(3,9),EXT(3),EUVWT(3,9),XS(3),
     1  DCS(3),S(3,3)
      COMMON/ERDCON/ERDI,XMUI,OMGI,CM(3)
      DIMENSION C(3),CC(3),D(9),DNO(3,3,40)
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
C    THIS SUBROUTINE IS DIMENSIONED FOR A MAXIMUM OF
C               40 OBSERVING STATIONS(MAXIMUM)
C               15 STATIONS OBSERVING ANY GIVEN PASS
C               10 ERROR MODEL UNKNOWNS FOR ANY PASS (FOR A TOTAL OF 16 UNKNOWNS
C                    FOR ANY PASS).
      DIMENSION ORBNAM(6),ORBUNK(16),DB(16)
      DIMENSION A(3),B(16)
      DIMENSION DN(3,3,40),BN(3,16,15),DK(3,40),DDN(16,16),DDK(16)
      INTEGER*2 LFLG(40),KSTATO(15),MODEL(10,2),IGUIDE(10,2)
      DIMENSION EMODEL(10,2)
      EQUIVALENCE(D(1),C(1)),(D(4),B(1))
      INTEGER*4 MODALF(3,2)
      DATA MODALF/'ZERO',' SET','    ','REFR','ACTI','ON  '/
      MAXSTA=40
      MAXSTO=15
      MAXEMU=10
      MAXUNK=16
      REWIND 1
      REWIND 3
      REWIND 4
      REWIND 2
      IF(PCODE(14).NE.0) NSTA=NSTA+1
C   INITIAL ACCUMULATING ARRAYS
      DO 60 KSTA=1,NSTA
      DO 50 J=1,3
      DO 40 I=1,3
      DNO(I,J,KSTA)=0.0
   40 DN(I,J,KSTA)=0.0
   50 DK(J,KSTA)=0.0
      VPVSTA(KSTA)=0.0
   60 NOBSTA(KSTA)=0
      WPW=0.0
      WPWSP=0.0
      NOBS=0
      NUNK=0
      READ(3)  ERDI,XMUI,OMGI,RO,VO
C    RO HOLDS A PRIORI COORDINATES OF CENTER OF MASS
C    VO HOLDS A PRIORI UNCERTAINTIES OF COORDINATES OF CENTER OF MASS
      DO 65 I=1,3
   65 CM(I)=RO(I)
```

```
      IF(PCODE(14).EQ.0) GO TO 75
      DO 70 I=1,3
C   UPDATE COORDINATES OF CENTER OF MASS
      CM(I)=STAUVW(I,NSTA)
C   IMPOSE A PRIORI CONSTRAINTS ON COORDINATES OF CENTER OF MASS
      IF(VO(I).EQ.0.0) GO TO 70
      DN(I,I,NSTA)=1.0/VO(I)**2
      DK(I,NSTA)=(RO(I)-STAUVW(I,NSTA))/VO(I)**2
   70 CONTINUE
   75 CONTINUE
C
  150 CONTINUE
C   PROCESS ANOTHER PASS
C   READ ORBIT HEADER
      READ(3) IORB,NORB,ORBNAM,EPOCH,IDAY,MONTH,IYR,GAST,RO,VO,TE,IH,MIN
     1,ESEC,XPM,YPM,CONTIN
      WRITE(1) IORB,NORB,ORBNAM,IDAY,MONTH,IYR,IH,MIN,ESEC,EPOCH,CONTIN
C  TEST FOR END OF DATA
      IF(CONTIN.EQ.ENDSIG)GO TO 700
      READ(4) ORBUNK,NSTATO,LFLG,KSTATO,NEMUNK,EMODEL,MODEL,IGUIDE
     1   , NUTORB
C   INITIALIZE ACCUMULATION ARRAYS
      WPWTO=0.0
      NOBSTO=0
      DO 170 KSTA=1,NSTATO
      DO 170 J=1,NUTORB
      DO 170 I=1,3
  170 BN(I,J,KSTA)=0.0
      DO 180 I=1,MAXUNK
      DDK(I)=0.0
      DO 180 J=1,MAXUNK
  180 DDN(I,J)=0.0
      WRITE(6,6001)      NORB,ORBNAM,IDAY,MONTH,IYR,IH,MIN,ESEC,EPOCH
 6001 FORMAT(/////////2X,A4,3X,6A8,' EPOCH=',I3,A3,1X,2I3,'H',I3,'M',
     1   F8.4,'S UT=MJD',F17.9)
      WRITE(6,6002) (ORBUNK(I),I=1,6)
 6002 FORMAT('0   CURRENT ORBIT ELEMENTS IN APPARENT SIDEREAL CARTESIAN
     1 COORDINATES'/5X,'POSITION(METERS)',4X,3F16.3/
     1    5X,'VELOCITY(METERS/SEC)',3F16.6)
      IF(NEMUNK.EQ.0) GO TO 185
      WRITE(6,6003)
 6003 FORMAT('0  CURRENT VALUE OF ERROR MODEL UNKNOWNS')
      DO 184 I=1,NEMUNK
      MODCOD=MODEL(I,1)
      KSTA=MODEL(I,2)
  184 WRITE(6,6004) (MODALF(J,MODCOD),J=1,3),KORDER(KSTA),(STANAM(J,KSTA
     1),J=1,5),ORBUNK(I+6)
 6004 FORMAT(5X,3A4,'FOR STATION',I6,2X,5A4,2X,'=',F15.3)
  185 CONTINUE
      WRITE(6,6005)
 6005 FORMAT('0STATION   DATE',T25,'TIME(UT)',3X,'CORRECTED RANGE',2X,
     1    'UNCERTAINTY',4X,'MISCLOSURE')
C   INITIALIZE ORBIT
      NFLG=1
      CALL ORBIT(NFLG,TE,TE,GAST,ORBUNK)
C
C   READ OBSERVATION RECORD
  200 READ (3) TOBS,TO,KSTA,GASTO,SINST,COSST,RSO,VARRA,ID,IDAY,MONTH,
```

```
     1   IYR,IH,MIN,SEC,S,CONTIN
      IF(CONTIN.EQ.ENDSIG) GO TO 350
      NN=1
      CALL DRIVER(NN,TOBS,GH,DT,XT,UVWT)
C   GET INTERNAL NUMBER OF THIS STATION IN THIS PASS
      ISTA=LFLG(KSTA)
C  TRANSFORM SATELLITE POSITION AND TRANSITION MATRIX TO TERRESTRIAL COORDINATES
      CALL DGMPRD(S,XT,EXT,3,3,1)
      CALL DGMPRD(S,UVWT,EUVWT,3,3,9)
COMPUTE STATION TO SATELLITE VECTOR
      DO 220 I=1,3
  220 XS(I)=EXT(I)-STAUVW(I,KSTA)
C   GET COMPUTED RANGE
      RSC=0.0
      DO 225 I=1,3
  225 RSC=RSC+XS(I)**2
      RSC=DSQRT(RSC)
C   GET STATION TO SATELLITE DIRECTION COSINES AND COMPUTE PARTIALS
      DO 230 I=1,3
      DCS(I)=XS(I)/RSC
      A(I)=-DCS(I)
  230 CONTINUE
      CALL DGMPRD(DCS,EUVWT,D,1,3,9)
C   PREPARE PARTIALS WITH RESPECT TO THE CENTER OF MASS.
      IF(PCODE(14).NE.0) CALL DGMPRD(S,C,CC,3,3,1)
C   FILL IN B WITH ZEROS
      IF(NUTORB.LE.6) GO TO 250
      DO 234 I=7,NUTORB
  234 B(I)=0.0
  235 CONTINUE
CORRECT OBSERVATION FOR ERROR MODEL
      IUNK=IGUIDE(ISTA,1)
      IF(IUNK.EQ.0) GO TO 245
      RSO=RSO-ORBUNK(IUNK)
      B(IUNK)=1.0
  245 IUNK=IGUIDE(ISTA,2)
      IF(IUNK.EQ.0) GO TO 250
COEFFICIENT OF REFRACTION TERM IS 1.0/(SIN OF ELEVATION ANGLE)
C   GET SIN OF ELEVATION ANGLE (SE)
C   GEODETIC LATITUDE IS APPROXIMATED BY SPHERICAL LATITUDE IN COMPUTATION
C               OF SIN OF ELEVATION ANGLE
      SE=0.0
      RSTA=0.0
      DO 246 I=1,3
      SE=SE+STAUVW(I,KSTA)*DCS(I)
  246 RSTA=RSTA+STAUVW(I,KSTA)**2
      SE=SE/DSQRT(RSTA)
      RSO=RSO-ORBUNK(IUNK)/SE
      B(IUNK)=1.0/SE
  250 CONTINUE
COMPUTE MISCLOSURE
      DL=RSO-RSC
      WRITE(6,6006)KORDER(KSTA),IDAY,MONTH,IYR,IH,MIN,SEC,RSO,VARRA,DL
 6006 FORMAT(I7,I4,1X,A3,I3,I5,I3,F8.4,F13.2,2F15.2)
      WT=1.0/VARRA**2
      WPWOBS=DL*WT*DL
      VPVSTA(KSTA)=VPVSTA(KSTA)+WPWOBS
      WPWTO=WPWTO+WPWOBS
```

```
      WPW=WPW+WPWOBS
      NOBSTA(KSTA)=NOBSTA(KSTA)+1
      NOBSTO=NOBSTO+1
      NOBS=NOBS+1
C
C  ACCUMULATE NORMAL EQUATIONS
      DO 270 I=1,3
      DK(I,KSTA)=DK(I,KSTA)+A(I)*WT*DL
      DO 260 J=1,3
  260 DN(I,J,KSTA)=DN(I,J,KSTA)+A(I)*WT*A(J)
      DO 265 J=1,NUTORB
  265 BN(I,J,ISTA)=BN(I,J,ISTA)+A(I)*WT*B(J)
  270 CONTINUE
      DO 280 I=1,NUTORB
      DDK(I)=DDK(I)+B(I)*WT*DL
      DO 280 J=1,NUTORB
  280 DDN(I,J)=DDN(I,J)+B(I)*WT*B(J)
      IF(PCODE(14).EQ.0) GO TO 200
C   ACCUMULATE NORMALS PERTAINING TO THE CENTER OF MASS
      DO 295 I=1,3
      DK(I,NSTA)=DK(I,NSTA)+CC(I)*WT*DL
      DO 291 J=1,3
      DN(I,J,NSTA)=DN(I,J,NSTA)+CC(I)*WT*CC(J)
  291 DNO(I,J,KSTA)=DNO(I,J,KSTA)+A(I)*WT*CC(J)
      DO 292 J=1,NUTORB
  292 BN(I,J,NSTATO)=BN(I,J,NSTATO)+CC(I)*WT*B(J)
  295 CONTINUE
C  RETURN TO PROCESS ANOTHER OBSERVATION
      GO TO 200
C
  350 CONTINUE
C   ENTER ON END OF PASS
      NUNK=NUNK+NUTORB
C  ADD A PRIORI CONSTRAINTS ON ORBIT UNKNOWNS TO DDN AND DDK AT THIS POINT
C  ADD A PRIORI CONSTRAINTS ON ERROR MODEL UNKNOWNS
      IF(NEMUNK.EQ.0) GO TO 370
      DO 369 I=1,NEMUNK
      IF(EMODEL(I,2).LE.0.0) GO TO 369
      DL=EMODEL(I,1)-ORBUNK(I+6)
      WT=1.0/EMODEL(I,2)**2
      DDN(I+6,I+6)=DDN(I+6,I+6)+WT
      DDK(I)=DDK(I)+WT*DL
      WPWOBS=DL*WT*DL
      NOBS=NOBS+1
      WPW=WPW+WPWOBS
  369 CONTINUE
  370 CONTINUE
      IF(NUTORB.EQ.MAXUNK)GO TO 380
C   PAD OUT DDN
      NN=NUTORB+1
      DO 379 I=NN,MAXUNK
      DDK(I)=0.0
  379 DDN(I,I)=1.0
C   INVERT DDN
      DET=1.0
  380 CALL DMINV(DDN,MAXUNK,DET,UVWT,B)
COMPUTE PARTIAL UNCERTAINTIES OF ORBIT UNKNOWNS
      DO 390 I=1,NUTORB
```

```
  390 B(I)=DSQRT(DDN(I,I))
      WRITE(6,6008) (B(I),I=1,NUTORB)
 6008 FORMAT(//' PARTIAL UNCERTAINTIES OF ORBIT UNKNOWNS-FROM DDN(INV) '
     1 ,10(/8D15.6))
C
      DO 395 I=1,NUTORB
      DO 395 J=1,NUTORB
  395 WPWSP=WPWSP+DDK(I)*DDN(I,J)*DDK(J)
      WRITE(1) DDN,DDK,BN
      WRITE(1) NUTORB,NSTATO,NEMUNK,KSTATO,LFLG,ORBUNK,EMODEL,MODEL,
     1 IGUIDE
      RMSTO=DSQRT(WPWTO/DFLOAT(NOBSTO))
      WRITE(6,6007) WPWTO,NOBSTO,RMSTO
 6007 FORMAT(//5X,'WEIGHTED SUM OF SQUARES OF MISCLOSURES =',F15.3/
     1  5X,'NUMBER OF OBSERVATIONS =',I8/5X,'RMS MISCLOSURE =',F15.3)
C  RETURN TO PROCESS ANOTHER PASS
      GO TO 150
C
C   ENTER HERE AT THE END OF ALL PASSES
  700 CONTINUE
      WRITE(2) DN,DK
      IF(PCODE(14).NE.0) WRITE(2) DNO
      REWIND 1
      REWIND 2
      REWIND 4
C  PERFORM ANALYSIS OF MISCLOSURES BY STATION
      WRITE(6,6019)
 6019 FORMAT(1H1,8(/),10X,'ANALYSIS OF MISCLOSURES BY STATION'//
     1T10,'STATION',T40,'NUMBER OF OBSERVATIONS',T70,'RMS MISCLOSURE')
      DO 750 KSTA=1,NSTA
      IF(NOBSTA(KSTA).GT.0) RMSMC=DSQRT(VPVSTA(KSTA)/DFLOAT(NOBSTA(KSTA)
     1))
      WRITE(6,6020) KORDER(KSTA),(STANAM(I,KSTA),I=1,5),NOBSTA(KSTA),
     1   RMSMC
 6020 FORMAT(T10,I7,1X,5A4,T55,I7,T70,F14.2)
  750 CONTINUE
COMPUTE DEGREES OF FREEDOM
      IDEGF=NOBS-NUNK
      WPW=WPW-WPWSP
      RMSMC=DSQRT(WPW/DFLOAT(IDEGF))
      WRITE(6,6021) NOBS,NUNK,IDEGF,WPW,RMSMC
 6021 FORMAT(////10X,'TOTAL NUMBER OF OBSERVATIONS'      ,T60,I8//
     110X,'TOTAL NUMBER OF ORBIT AND ERROR MODEL UNKNOWNS',T60,I8//
     210X,'CORRESPONDING DEGREES OF FREEDOM',T60,I8//
     310X,'TOTAL SUM OF SQUARES OF MISCLOSURES, I.E.,VPV+XU',T60,F11.2//
     410X,'CORRESPONDING STANDARD DEVIATION OF UNIT WEIGHT',T60,F11.2)
      RETURN
      END
```

```
      SUBROUTINE POLE(DT, XPM,YPM)
      DOUBLE PRECISION DT,XPM,YPM
      DIMENSION PM(286,2),PMT(2)
      DIMENSION PMX1(90),PMY1(90),PMX2(90),PMY2(90),PMX3(15),PMY3(15)
      EQUIVALENCE (PM,PMX1),(PM(1,2),PMY1),(PM(91,1),PMX2),
     1(PM(91,2),PMY2),(PM(181,1),PMX3),(PM(181,2),PMY3)
      DIMENSION PMX4(5),PMY4(5)
      EQUIVALENCE (PM(196,1),PMX4),(PM(196,2),PMY4)
      DIMENSION PMX5(86),PMY5(86)
      EQUIVALENCE (PM(201,1),PMX5),(PM(201,2),PMY5)
C    POLAR MOTION TABLES FURNISHED BY TOMLINSON (TAKEN FROM IPMS)
      DATA PMX1/
     1-0.173,-0.215,-0.235,-0.237,-0.218,-0.162,-0.097,-0.032, 0.036,
     2 0.111, 0.188, 0.237, 0.348, 0.398, 0.398, 0.368, 0.330, 0.280,
     3 0.218, 0.144, 0.069, 0.000,-0.062,-0.112,-0.140,-0.153,-0.151,
     4-0.126,-0.086,-0.037, 0.026, 0.092, 0.161, 0.223, 0.272, 0.299,
     5 0.308, 0.296, 0.261, 0.202, 0.135, 0.073, 0.046, 0.035, 0.013,
     6-0.026,-0.072,-0.096,-0.107,-0.103,-0.087,-0.039, 0.004, 0.040,
     7 0.070, 0.080, 0.109, 0.117, 0.117, 0.109, 0.092, 0.074, 0.065,
     8 0.064, 0.062, 0.057, 0.046, 0.034, 0.030, 0.032, 0.040, 0.043,
     9 0.042, 0.041, 0.039, 0.028, 0.019,-0.010,-0.027,-0.021,-0.009,
     A 0.008, 0.027, 0.047, 0.071, 0.095, 0.120, 0.144, 0.162, 0.173  /
      DATA PMX2/
     1 0.171, 0.157, 0.128, 0.094, 0.056, 0.017,-0.019,-0.054,-0.086,
     2-0.110,-0.121,-0.119,-0.105,-0.076,-0.038, 0.009, 0.070, 0.134,
     3 0.191, 0.239, 0.274, 0.301, 0.281, 0.237, 0.176, 0.112, 0.048,
     4-0.011,-0.069,-0.122,-0.171,-0.206,-0.194,-0.169,-0.139,-0.101,
     5-0.055, 0.004, 0.074, 0.164, 0.214, 0.240, 0.241, 0.239, 0.255,
     6 0.250, 0.219, 0.161, 0.099, 0.042,-0.012,-0.067,-0.120,-0.160,
     7-0.185,-0.196,-0.194,-0.174,-0.130,-0.072,-0.003, 0.071, 0.127,
     8 0.168, 0.201, 0.221, 0.227, 0.220, 0.194, 0.138, 0.075, 0.033,
     9 0.000,-0.029,-0.058,-0.086,-0.105,-0.116,-0.119,-0.115,-0.104,
     A-0.086,-0.057,-0.010, 0.052, 0.096, 0.117, 0.125, 0.123, 0.115/
      DATA PMY1/
     1 0.022, 0.098, 0.187, 0.265, 0.328, 0.389, 0.443, 0.478, 0.493,
     2 0.478, 0.447, 0.411, 0.365, 0.307, 0.235, 0.165, 0.097, 0.043,
     3-0.007,-0.038,-0.057,-0.064,-0.057,-0.025, 0.032, 0.120, 0.211,
     4 0.285, 0.340, 0.372, 0.393, 0.406, 0.410, 0.401, 0.370, 0.320,
     5 0.260, 0.201, 0.143, 0.090, 0.043, 0.007,-0.012,-0.007, 0.025,
     6 0.059, 0.094, 0.123, 0.153, 0.182, 0.209, 0.238, 0.263, 0.288,
     7 0.300, 0.306, 0.301, 0.288, 0.271, 0.249, 0.220, 0.189, 0.161,
     8 0.150, 0.151, 0.158, 0.161, 0.160, 0.155, 0.153, 0.150, 0.151,
     9 0.154, 0.157, 0.165, 0.174, 0.191, 0.212, 0.242, 0.276, 0.297,
     A 0.309, 0.314, 0.312, 0.304, 0.290, 0.271, 0.246, 0.214, 0.175/
      DATA PMY2/
     1 0.132, 0.092, 0.068, 0.060, 0.067, 0.083, 0.104, 0.128, 0.160,
     2 0.200, 0.248, 0.295, 0.329, 0.356, 0.376, 0.388, 0.387, 0.375,
     3 0.349, 0.307, 0.251, 0.193, 0.139, 0.091, 0.046, 0.008,-0.020,
     4 0.005, 0.041, 0.078, 0.120, 0.168, 0.230, 0.294, 0.353, 0.412,
     5 0.455, 0.467, 0.459, 0.436, 0.394, 0.339, 0.275, 0.219, 0.168,
     6 0.123, 0.085, 0.060, 0.046, 0.043, 0.049, 0.069, 0.103, 0.153,
     7 0.226, 0.286, 0.334, 0.374, 0.408, 0.434, 0.444, 0.433, 0.399,
     8 0.349, 0.303, 0.259, 0.221, 0.186, 0.156, 0.131, 0.114, 0.103,
     9 0.098, 0.100, 0.108, 0.124, 0.149, 0.181, 0.215, 0.255, 0.298,
     A 0.330, 0.344, 0.345, 0.337, 0.324, 0.308, 0.291, 0.273, 0.253/
      DATA PMX3/
     1 0.099, 0.079, 0.056, 0.031, 0.012,-0.001,-0.006,-0.008,-0.002,
     2 0.012, 0.035, 0.055, 0.046, 0.027, 0.008  /
```

```
      DATA PMX4              /-0.010,-0.029,-0.049,-0.063,-0.066/
      DATA PMY3/
     1 0.233, 0.213, 0.194, 0.177, 0.165, 0.157, 0.155, 0.154, 0.152,
     2 0.156, 0.163, 0.172, 0.183, 0.195, 0.208  /
      DATA PMY4              / 0.220, 0.234, 0.249, 0.269, 0.289/
      DATA PMX5/
     1-.056,-.037,-.014,.008,.031,.051,.064,.067,.064,.060,.088,.119,.10
     26 ,.054,.008,-.027,-.056,     -.084,-.109,-.123,-.127,-.120,-.102
     3,-.073,-.033,.010,.052,.091,.125,.154,.174,.185,.184,.168,.127,.07
     47 ,.029,-.021,-.071,-.115,-.157,-.184,-.184,-.166,-.135,-.100,-.06
     53 ,-.025,.017,.083,.154,.210,.241,.250,.238,.207,.167,.119,.068,
     6.019,-.045,-.127,-.211,-.242,-.225,-.184,-.135,-.078,-.027,.026,
     7.086,.150,.214,.261,.270,.256,.220,.177,.143,.114,.071,.015,-.015,
     8-.042,-.081,-.125/
      DATA PMY5/
     1.302,.308,.308,.302,.290,.276,.260,.245,.231,.216,.202,.183,.166,
     2.157,.156,.161,.172,     .197,.233,.265,.289,.310,.330,.350,.370,
     3.386,.392,.386,.367,.337,.302,.260,.212,.167,.134,.115,.105,.104,
     4.114,.134,.168,.216,.273,.333,.384,.419,.449,.465,.463,.436,.391,
     5.347,.303,.252,.196,.148,.112,.080,.045,.017,.020,.052,.097,.149,
     6.204,.270,.340,.389,.443,.478,.482,.468,.444,.409,.337,.276,.236,
     7.160,.166,.122,.070,.033,.030,.053,.109,.174/
      A=(DT-0.36203861D5)*0.54758185D-1
      L=A+1.0
      IF(L.LT.2)GOTO901
      IF(L.GT.284) GO TO 901
      TL=L
      AN=A+1.0-TL
      B=AN*(AN-1.0)/4.0
      DO10I=1,2
      DEL0=PM(L,I)-PM(L-1,I)
      DEL1=PM(L+1,I)-PM(L,I)
      DEL2=PM(L+2,I)-PM(L+1,I)
      PMT(I)=PM(L,I)+AN*DEL1+B*(DEL2-DEL0)
   10 CONTINUE
      XPM=PMT(1)
      YPM=PMT(2)
      RETURN
  901 WRITE(6,9001)
 9001 FORMAT(72H0TABLES OF POLAR MOTION COVER ONLY FROM 1958.0 TO 1972.2
     15, PLEASE EXTEND)
      STOP
      END
```

```
      SUBROUTINE KEPTCE(ORBEL,Z)
CONVERT FROM KEPLERIAN TO CARTESIAN ORBIT ELEMENTS
C             (EPOCH POSITION AND VELOCITY VECTORS)
      IMPLICIT REAL*8(A-Z)
      COMMON/ERDCON/AE,GM,OMGI,XCM,YCM,ZCM
      DIMENSION ORBEL(6),Z(6),RXQ(3,3),Q(3,2)
      DATA RHOD/   57.295779513082/
      A=ORBEL(1)
      ECC=ORBEL(2)
      INC=ORBEL(3)/RHOD
      NODE=ORBEL(4)/RHOD
      ARGP=ORBEL(5)/RHOD
      M=ORBEL(6)/RHOD
C
      CALL KEPEQ(M,ECC,1.0D-12,E)
      CE=DCOS(E)
      SE=DSIN(E)
      Q(1,1)=A*(CE-ECC)
      Q(2,1)=A*DSQRT(1.0-ECC**2)*SE
      Q(3,1)=0.0
      N=DSQRT(GM/A**3)
      FACTOR=N*A/(1.0-ECC*CE)
      Q(1,2)=-SE*FACTOR
      Q(2,2)=DSQRT(1.0-ECC**2)*CE*FACTOR
      Q(3,2)=0.0
      CN=DCOS(NODE)
      SN=DSIN(NODE)
      CW=DCOS(ARGP)
      SW=DSIN(ARGP)
      CI=DCOS(INC)
      SI=DSIN(INC)
      RXQ(1,1)=CN*CW-SN*CI*SW
      RXQ(1,2)=-CN*SW-SN*CI*CW
      RXQ(1,3)=SN*SI
      RXQ(2,1)= SN*CW+SW*CN*CI
      RXQ(2,2)=-SN*SW+CW*CN*CI
      RXQ(2,3)=-CN*SI
      RXQ(3,2)=CW*SI
      RXQ(3,1)=SW*SI
      RXQ(3,3)=CI
      CALL DGMPRD(RXQ,Q(1,1),Z(1),3,3,1)
      CALL DGMPRD(RXQ,Q(1,2),Z(4),3,3,1)
      RETURN
      END
```

```
      SUBROUTINE KEPEQ(M,EC,CONV,E)
      IMPLICIT REAL*8(A-H,K-Z)
      E=M+EC*DSIN(M)
      DO 10 I=1,50
      DELE=(M-E+EC*DSIN(E))/(1.0-EC*DCOS(E))
      E=E+DELE
      IF(DABS(DELE).LT.CONV) GO TO20
   10 CONTINUE
      WRITE(6,100)
  100 FORMAT(52HKEPLERS EQUATION FAILS TO CONVERGE IN 50 ITERATIONS    )
   20 RETURN
      END
```

```
      SUBROUTINE ORBIT(NFLG,TON,TEN,ALFI,XIN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION
     1 XPO(10),YPO(10),ZPO(10),UMT(224),VMT(224),UVM(180),XIN(6)
     2,CNM(15),SNM(15),CTT(15),CTB(15),LCT(6),ICT(8),XM(6)
      DIMENSION CMC(3),CMI(3)
      COMMON/ORBPAR/TZE,STEP,ALF,OMG,EPS,CUD,CUV,CUT,XM,XPO,YPO,ZPO,UVM
     1,CMC
      COMMON/ERDCON/ERDI,XMUI,OMGI,XCM,YCM,ZCM
      EQUIVALENCE(CMI(1),XCM)
C     VALUES OF SPHERICAL HARMONIC COEFFICIENTS FROM SAO 1969 STANDARD EARTH
C           C20.5     FIELD      SAO        1969
      DATA CNM/1.0D0,0.0D0,-.1082628D-02,0.25380D-05,0.15930D-05,2*0.D0,
     1  0.21276D-5,-.50270D-06,0.15575D-05,.30469D-06,.73844D-07,
     2  .95700D-07,.59130D-07,-.16838D-08/
      DATA SNM/7*0.0D0,.28099D-06,-.46263D-06,-.88052D-06,-.21678D-06,
     1     .15794D-06,.19946D-06,-.92433D-08,.71686D-08/
C     ARRAYS OF CONSTANT DEPENDING ON DEGREE AT WHICH GRAVITY FIELD IS TRUNCATED
      DATA LCT/5,4,3,2,1,0/
      DATA CTT/-2.0D0,-4.0D0,-6.0D0,-8.0D0,-10.0D0,-2.0D0,-4.0D0,
     1  -6.0D0,-8.0D0,-2.0D0,-4.0D0,-6.0D0,-2.0D0,-4.0D0,-2.0D0/
      DATA CTB/2.0D0,6.0D0,12.0D0,20.0D0,30.0D0,2.0D0,6.0D0,12.0D0,
     1  20.0D0,2.0D0,6.0D0,12.0D0,2.0D0,6.0D0,2.0D0/
      CNM(2)=ZCM/ERDI
      CNM(6)=XCM/ERDI
      SNM(6)=YCM/ERDI
CCC    USE THIS VALUE FOR COMPARSION PURPOSES ONLY
      CNM(5)=0.15903E-05
      KFLG=1
      NTE=5
      NHT=0
      KRG=0
      KDR=0
      KV=0
      KT=10
      KTR=KT
C    NFLG=0 MEANS TO COMPUTE THE CONDITIONS AT TEN FROM TON AND
C        RETURN THE NEW POSITION AND VELOCITY ELEMENTS IN XIN
C    NFLG=1 MEANS TO UPDATE THE ORBIT FROM TON TO TEN IF NECESSARY AND TO
C       INITIALIZE THE SERIES COEFFICIENTS AT TON
C    NFLG=2 INDICATES THAT THE EXPANSIONS ALREADY EXIST FOR THIS ORBIT,
C         BUT THE ORBIT IS TO BE INTEGRATED UP TO TEN AND NEW EXPANSIONS
C           FORMED ABOUT THAT POINT
      IF(NFLG-1) 150,120,240
  120 DO 130 I=1,36
  130 UVM(I)=0.0
      DO 140 I=1,9,4
      UVM(I+27)=1.0
  140 UVM(I)=1.0
  150 CONTINUE
   10 CONTINUE
C
COMPUTE CONONICAL UNITS AND INITIALIZE CONSTANTS
      CUM=XIN(1)*XIN(1)+XIN(2)*XIN(2)+XIN(3)*XIN(3)
      CUD=DSQRT(CUM)
      CUM=CUM*CUD
      CUT=DSQRT(CUM/XMUI)
      CUV=CUD/CUT
```

```
      ERD=ERDI/CUD
      XMU=1.0
      ALF=ALFI
      OMG=OMGI*CUT
      EPS=0.04/CUD
      XPO(1)=XIN(1)/CUD
      YPO(1)=XIN(2)/CUD
      ZPO(1)=XIN(3)/CUD
      XPO(2)=XIN(4)/CUV
      YPO(2)=XIN(5)/CUV
      ZPO(2)=XIN(6)/CUV
      TIN=TON/CUT
      TFI=TEN/CUT
      DO 180 I=1,3
  180 CMC(I)=CMI(I)/CUD
  200 CONTINUE
      TZE=TIN
      DELT=TFI-TZE
      IF(DABS(DELT).GT.0.0) GO TO 220
      KFLG=2
      IF(NFLG.NE.0) GO TO 220
C       SET UP RETURN FOR NFLG=0
      XIN(1)=XPO(1)*CUD
      XIN(2)=YPO(1)*CUD
      XIN(3)=ZPO(1)*CUD
      XIN(4)=XPO(2)*CUV
      XIN(5)=YPO(2)*CUV
      XIN(6)=ZPO(2)*CUV
      RETURN
  220 CONTINUE
      KEY=KFLG
      CALL EXPAND       (XPO,YPO,ZPO,CNM,SNM,LCT,ICT,UMT,VMT,CTB,CTT
     1,ERD,XMU,ALF,OMG,ECC,NTE,KTR,KDR,NHT,CDC,CTW,KEY,DMT,KRG,CMC)
      ICN=1
      CALL UPDATE (ICN,KTR,EPS,STEP,XPO,YPO,ZPO,XM(1),XM(3),XM(5))
      IF(KFLG.NE.2) GO TO 250
      CALL VARIEQ(XPO,YPO,ZPO,CNM,SNM,LCT,ICT,UMT,VMT,UVM,ERD,ALF,OMG,CD
     1C,CTW,NTE,KTR,KDR)
C     WRITE(6,356) TFI,STEP,ALF,OMG,EPS,CUD,CUV,CUT,XPO,YPO,ZPO,UVM
  356 FORMAT(2(4E20.12/),6(5E20.12/),30(6E20.12/))
      RETURN
C     ENTER HERE WITH NFLG=2
  240 TIN=TON
      TFI=TEN
      TZE=TIN
      DELT=TFI-TZE
C
C     INTEGRATE ORBIT
  250 DEL=DELT
      IF(DABS(DELT).GT.STEP) DEL=DSIGN(STEP,DELT)
      ICN=2
      CALL UPDATE(ICN,KTR,EPS,DEL,XPO,YPO,ZPO,XM(1),XM(3),XM(5))
      TIN=TIN+DEL
      ALF=ALF+DEL*OMG
C     WRITE(6,356) TIN,DEL,ALF,OMG,EPS,CUD,CUV,CUT,XPO,YPO,ZPO,XM
      XPO(1)=XM(1)
      YPO(1)=XM(3)
      ZPO(1)=XM(5)
```

```
XPO(2)=XM(2)
YPO(2)=XM(4)
ZPO(2)=XM(6)
GO TO 200
END
```

```
      SUBROUTINECLEAR(A,N)
      IMPLICITREAL*8(A-H,O-Z)
C
C     FILL  A  ARRAY WITH FLOATING POINT ZEROES.
C
C
C
      DIMENSIONA(1)
C
      DO10J=1,N
   10 A(J)=0.
      RETURN
      END
```

```
      SUBROUTINE DRIVER (NN,TM,GH,DT,XOT,VEM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XOT(1),VEM(1)
      DIMENSION XM(6),XPO(10),YPO(10),ZPO(10),UVM(180)
      DIMENSION AMT(36)
      COMMON /ORBPAR/TEP,DLT,ALF,OMG,EPS,CUD,CUV,CUT,XM,XPO,YPO,ZPO,UVM,
     1 CMC(3)
   50 TOT=TM/CUT
   51 DEL=TOT-TEP
      NFLG=1
      IF(DABS(DEL).LT.DLT) GO TO 67
C   TIME FROM T ZERO IS TOO LARGE FOR CONVERGENCE OF SERIES
C   INCREMENT T ZERO BY ONE STEP
      DEL=DSIGN(DLT,DEL)
      NFLG=2
   67 CONTINUE
      KTR=9
      CALL MATRUP(KTR,DEL,UVM,AMT)
      IF(NFLG.EQ.1) GO TO 69
      DO 68 I=1,36
   68 UVM(I)=AMT(I)
      TON=TEP
      TEN=TEP+DEL
      ALFI=ALF
      CALL ORBIT(NFLG,TON,TEN,ALFI,XOT)
      GO TO 51
   69 CONTINUE
      GH=ALF+DEL*OMG
      KTR=10
      ICN=2
  100 FORMAT(6E18.5)
      CALL UPDATE (ICN,KTR,EPS,DEL,XPO,YPO,ZPO,XM(1),XM(3),XM(5))
      XOT(1)=XM(1)*CUD
      XOT(2)=XM(3)*CUD
      XOT(3)=XM(5)*CUD
      XOT(4)=XM(2)*CUV
      XOT(5)=XM(4)*CUV
      XOT(6)=XM(6)*CUV
      J=10
      K=19
      DO 70 I=1,9
      VEM(J)=AMT(I)
      VEM(K)=AMT(J)*CUT
      VEM(I)=-VEM(J)
      J=J+1
      K=K+1
   70 CONTINUE
      DO 80 I=1,9,4
   80 VEM(I)=VEM(I)+1.0
      RETURN
      END
```

```
      SUBROUTINEUPDATE(ICN,KTR,EPS,DEL,XPO,YPO,ZPO,XOT,YOT,ZOT)
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONXPO(1),YPO(1),ZPO(1),XOT(1),YOT(1),ZOT(1)
C   ICN=1 - COMPUTE STEP (DEL)      KTR - NUMBER OF TERMS TO USE
C   ICN=2 - INTERGRATE POS.,VEL.    EPS - TRUNCATION ERROR LIMIT
C   ICN=3 - DO BOTH TOGETHER        DEL - VALID STEP SIZE
C     INITIAL AND FINAL ADDRESSES FOR X,Y,Z MAY BE SYNONYMOUS.
C
      KA=KTR
      KB=ICN-2
      IF(KB)10,20,10
   10 CONTINUE
C   SCALE DATA TO REMAIN IN COMPUTER RANGE
      B=1.0
      A=DSQRT((B*XPO(KA))**2+(B*YPO(KA))**2+(B*ZPO(KA))**2)
      A=DLOG((B*EPS)/A)
      B=KA-1
      DEL=DEXP(A/B)
      IF(KB)40,20,20
   20 XPI=XPO(KA)
      YPI=YPO(KA)
      ZPI=ZPO(KA)
      KA=KA-1
      KB=KA
      A=KA
      XVI=XPI*A
      YVI=YPI*A
      ZVI=ZPI*A
      DT=DEL
      DO30I=2,KB
      XPI=XPI*DT+XPO(KA)
      YPI=YPI*DT+YPO(KA)
      ZPI=ZPI*DT+ZPO(KA)
      A=KA-1
      XVI=XVI*DT+A*XPO(KA)
      YVI=YVI*DT+A*YPO(KA)
      ZVI=ZVI*DT+A*ZPO(KA)
      KA=KA-1
   30 CONTINUE
      XOT(1)=XPI*DT+XPO(1)
      YOT(1)=YPI*DT+YPO(1)
      ZOT(1)=ZPI*DT+ZPO(1)
      XOT(2)=XVI
      YOT(2)=YVI
      ZOT(2)=ZVI
   40 RETURN
      END
```

```
      SUBROUTINEMATRUP(KTR,DEL,UVM,TAR)
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONTAR(36),UVM(1),UVO(1)
C   SUBROUTINE TO UPDATE THE MATRIZANT WITH RESPECT TO TIME (DEL).
C
C
      L=KTR
      N=18*(L-1)+1
      DO10I=1,18
      TAR(I)=UVM(N)
      TAR(I+18)=0.0
   10 N=N+1
      DO20I=2,L
      N=18*(L-I)+1
      M=N+18
      TM=L-I+1
      DO20K=1,18
      TAR(K)=DEL*TAR(K)+UVM(N)
      TAR(K+18)=DEL*TAR(K+18)+TM*UVM(M)
      N=N+1
      M=M+1
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROT3(ANG,X)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(3),XX(3)
      COS=DCOS(ANG)
      SIN=DSIN(ANG)
      XX(1)=COS*X(1)+SIN*X(2)
      XX(2)=-SIN*X(1)+COS*X(2)
      DO 100 I=1,2
  100 X(I)=XX(I)
      RETURN
      END
```

```
      SUBROUTINE ORBRN
C  FORM REDUCED NORMAL EQUATIONS FOR UP TO 40 STATIONS
C   FOR SHORT ARC MODE PROCESSING
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/NSTA/NSTA
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      COMMON/WPW/WPW,XPU,IDEGF,IFSTA
      INTEGER CONTIN,ENDSIG/1HE/
      COMMON/STAORD/KORDER(150)
      COMMON/NORMEQ/REDN(3,3,820),U(3,40),L(820),LSOLVE
      DIMENSION BN(3,16,15),DUMMY(3,3,40),DDN(16,16),DDK(16),BNDDNI(3,
     1  16),DDK3(3)
      INTEGER*2 L,LSOLVE,LG(40),IDAY,IYR,KSTATO(15)
      DIMENSION ORBNAM(6)
      EQUIVALENCE(BN(1,1,1),DUMMY(1,1,1)),(DDK3(1),DDK(1))
      LOC(K)=(K*(K+1))/2
      MAXSTA=40
      IF(NSTA.GT.MAXSTA) GO TO 901
C
C   THE REDUCED NORMAL EQUATIONS ARE STORED AS 3 X 3 BLCCKS IN THE ARRAY REDN.
C   ONLY THE UPPER TRIANGULAR PART OF THE REDUCED NORMAL EQUATIONS IS STORED.
C     THE BLOCKS OF THE REDUCED NORMAL EQUATIONS ARE NUMBERED
C       ACCORDING TO THE FOLLOWING SCHEME:
C
C         1     2     4     7    11
C               3     5     8    12
C                     6     9    13
C                          10    14
C                                15            ET CETERA
C
C  L(820) IS THE GUIDE MATRIX
C  L=1  SIGNIFIES A NON ZERO BLOCK
C  L=0  SIGNIFIES A ZERO BLOCK
      REWIND 1
      REWIND 2
      IB=LOC(NSTA)
      DO 100 JB=1,IB
      DO 99 I=1,3
      DO 99 J=1,3
   99 REDN(I,J,JB)=0.0
  100 L(JB)=0
C
      READ (2) DUMMY,U
C
C  STASH  DIAGONAL BLOCKS
      DO 110 KSTA=1,NSTA
      IB   =LOC(KSTA)
      DO 108 I=1,3
      DO 108 J=1,3
  108 REDN(I,J,IB)=DUMMY(I,J,KSTA)
  110 CONTINUE
C
      IF(PCODE(14).EQ.0) GO TO 130
      READ(2) DUMMY
      NN=NSTA-1
      IB=LOC(NN)
      DO 120 KSTA=1,NN
```

```
      NB=IB+KSTA
      DO 120 I=1,3
      DO 120 J=1,3
  120 REDN(I,J,NB)=DUMMY(I,J,KSTA)
  130 CONTINUE
C
      FDEGF=IDEGF
      IF(PCODE(9).EQ.1) WRITE(7,7010) FDEGF,WPW
 7010 FORMAT(16X,2F16.6)
C   READ BLOCKS FROM EACH ORBIT AND REDUCE NORMAL EQUATIONS.
C
  150 READ (1) IORB,NORB,ORBNAM,IDAY,MONTH,IYR,IH,MIN,ESEC,EPOCH,CONTIN
C    IF END OF DATA, GO OUT OF LOOP
      IF(CONTIN.EQ.ENDSIG) GO TO 400
      READ (1) DDN,DDK,BN
      READ (1) NUTORB,NSTATO,NEMUNK,KSTATO
      DO 180 IS=1,NSTATO
      ISTA=KSTATO(IS)
C   GET BN * DDN(INVERSE)
      DO 160 I=1,3
      DO 160 J=1,NUTORB
      BNDDNI(I,J)=0.0
      DO 160 K=1,NUTORB
  160 BNDDNI(I,J)=BNDDNI(I,J)+BN(I,K,IS)*DDN(K,J)
      DO 165 I=1,3
      DO 165 K=1,NUTORB
  165 U(I,ISTA)=U(I,ISTA)-BNDDNI(I,K)*DDK(K)
      DO 180 JS=1,NSTATO
      JSTA=KSTATO(JS)
C    SKIP IF (ISTA.GT.JSTA), SINCE ONLY THE UPPER TRIANGULAR PART OF THE
C        REDUCED NORMAL EQUATIONS IS BEING COMPUTED AND SAVED.
      IF(ISTA.GT.JSTA) GO TO 180
      NB=LOC(JSTA-1)+ISTA
      DO 170 I=1,3
      DO 170 J=1,3
      DO 170 K=1,NUTORB
  170 REDN(I,J,NB)=REDN(I,J,NB)-BNDDNI(I,K)*BN(J,K,JS)
      L(NB)=L(NB)+1
  180 CONTINUE
C    RETURN TO PROCESS ANOTHER PASS
      GO TO 150
C
C    ENTER HERE WHEN ALL PASSES HAVE BEEN PROCESSED
  400 CONTINUE
C
C   SIMULATE KRAKIWSKI'S GUIDE MATRIX
      IF(PCODE(6).NE.1) GO TO 441
C
      WRITE(6,6001)
 6001 FORMAT(1H1,10(/),20X,'GUIDE MATRIX')
      DO 440 ISTA=1,NSTA
      IB=0
      LG(1)=1000
      DO 435 JSTA=ISTA,NSTA
      JB=LOC(JSTA-1)+ISTA
      IF(L(JB).EQ.0) GO TO 435
      IB=IB+1
      LG(IB)=KORDER(JSTA)
```

```
  435 CONTINUE
C
      IB=IB+1
      IF(IB.GT.1) LG(IB)=999
  439 WRITE(6,6002) KORDER(ISTA),(LG(I),I=1,IB)
 6002 FORMAT(20X,I5,5X,18I5,200(/30X,18I5))
  440 CONTINUE
  441 CONTINUE
C
C  PRINT NORMALS IN ASD FORMAT,AND PUNCH IF DESIRED.
      WRITE(6,6003)
 6003 FORMAT(1H1//'                NORMAL EQUATIONS (SEE GUIDE MATRIX)'//)
      DO 450 ISTA=1,NSTA
      DO 442 I=1,3
  442 DDK(I)=-U(I,ISTA)
      IB=0
      JB=LOC(ISTA)
      IF(L(JB).GT.0) IB=1
C   PUNCH NORMALS
      IF(PCODE(9).NE.1) GO TO 443
      WRITE(7,7001) KORDER(ISTA)
 7001 FORMAT(14I5)
      WRITE(7,7006) DDK3
 7006 FORMAT(3(F16.10,5X))
      WRITE(7,7008) ((REDN(I,J,JB),J=1,3),I=1,3)
 7008 FORMAT(3F16.10/3F16.10/3F16.10)
C
  443 CONTINUE
C   PRINT DIAGONAL BLOCK
      IF(PCODE(7).NE.1) GO TO 444
      WRITE(6,6004) KORDER(ISTA)
 6004 FORMAT(//I5)
      WRITE(6,6006) DDK3
 6006 FORMAT(/3(F16.10,5X))
      WRITE(6,6008) ((REDN(I,J,JB),J=1,3),I=1,3)
 6008 FORMAT(3F16.10)
  444 CONTINUE
C   PRINT OFF-DIAGONAL BLOCKS
      KSTA=ISTA+1
      IF(ISTA.EQ.NSTA) GO TO 448
      DO 445 JSTA=KSTA,NSTA
      JB=LOC(JSTA-1)+ISTA
      IF(L(JB).EQ.0) GO TO 445
      IB=IB+1
      IF(PCODE(9).NE.1) GO TO 7445
      WRITE(7,7001) KORDER(JSTA)
      WRITE(7,7008) ((REDN(I,J,JB),J=1,3),I=1,3)
 7445 CONTINUE
      IF(PCODE(7).NE.1) GO TO 445
      WRITE(6,6004) KORDER(JSTA)
      WRITE(6,6008) ((REDN(I,J,JB),J=1,3),I=1,3)
  445 CONTINUE
  448 I=1000
      IF(IB.GT.0) I=999
      IF(PCODE(7).EQ.1) WRITE(6,6004) I
      IF(PCODE(9).EQ.1) WRITE(7,7001) I
  450 CONTINUE
      IF(PCODE(8).NE.1) GO TO 478
```

```
      WRITE(6,6010)
 6010 FORMAT(10(/),20X,'OBSERVATIONS ON EACH LINE')
      IB=NSTA-1
      DO 475 ISTA=1,IB
      KSTA=ISTA+1
      DO 475 JSTA=KSTA,NSTA
      WRITE(6,6011) KORDER(ISTA),KORDER(JSTA),L(LOC(JSTA-1)+ISTA)
 6011 FORMAT(8I10)
  475 CONTINUE
  478 CONTINUE
      RETURN
  901 CONTINUE
      WRITE(6,9001) MAXSTA,NSTA
 9001 FORMAT('    FORMRN IS PRESENTLY DIMENSIONED TO HANDLE ONLY',I5,
     1'    UNKNOWN STATIONS.'/20X,' THIS PROBLEM HAS',I5,'    UNKNOWN STAT
     2IONS.'/10X,'EXECUTION IS TERMINATED BY PROGRAM.')
      STOP
      END
```

```
      SUBROUTINE PORB
C     PRINT UPDATED ORBIT ELEMENTS AND ERROR MODEL TERMS
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/NSTA/NSTA
      COMMON/WPW/WPW,XPU,IDEGF,NFSTA
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      INTEGER STANAM,IDS*2
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/STAORD/KORDER(150)
      DIMENSION DX(3,40), DDN(16,16),DDK(16),BN(3,16,15),DDKM(16)
      INTEGER CONTIN,ENDSIG/1HE/
      INTEGER*2 LFLG(40),KSTATO(15),MODEL(10,2),IGUIDE(10,2)
      INTEGER*2 IDAY,IYR
      DIMENSION EMODEL(10,2),ORBNAM(6),ORBUNK(16),DB(16)
      DIMENSION COVX(3,3)
      INTEGER*4 MODALF(3,2)
      DATA MODALF/'ZERO',' SET','    ','REFR','ACTI','ON  '/
      REWIND 1
      REWIND 2
      REWIND 4
      MAXUNK=16
      WRITE(6,6001)
 6001 FORMAT(1H1,/////10X,'CORRECTIONS TO ORBIT AND ERROR MODEL UNKNOWNS
     1')
C     SKIP HEADER RECORD ON 2
      READ(2)
C     GET CORRECTIONS AT EACH STATION
      DO 120 ISTA=1,NSTA
  120 READ(2)(DX(I,ISTA),I=1,3),COVX
      REWIND 2
C     BEGIN PROCESSING PASSES
  150 CONTINUE
C READ ORBIT HEADER
      READ (1) IORB,NORB,ORBNAM,IDAY,MONTH,IYR,IH,MIN,ESEC,EPOCH,CONTIN
      IF(CONTIN.EQ.ENDSIG) GO TO 700
      READ (1) DDN,DDK,BN
C     GET PREVIOUS SET OF UNKNOWNS
      READ (1) NUTORB,NSTATO,NEMUNK,KSTATO,LFLG,ORBUNK,EMODEL,MODEL,
     1 IGUIDE
      WRITE(6,6002)      NORB,ORBNAM,IDAY,MONTH,IYR,IH,MIN,ESEC,EPOCH
 6002 FORMAT(////////2X,A4,3X,6A8,' EPOCH=',I3,A3,1X,2I3,'H',I3,'M',
     1   F8.4,'S UT=MJD',F17.9)
COPY DDK INTO DDKM
      DO 170 I=1,MAXUNK
  170 DDKM(I)=DDK(I)
      DO 180 IS=1,NSTATO
      ISTA=KSTATO(IS)
      DO 180 I=1,NUTORB
      DO 180 J=1,3
  180 DDKM(I)=DDKM(I)-BN(J,I,IS)*DX(J,ISTA)
      CALL DGMPRD(DDN,DDKM,DB,MAXUNK,MAXUNK,1)
      WRITE(6,6003)(DB(I),I=1,NUTORB)
 6003 FORMAT(' CORRECTION VECTOR',100(/6F20.8))
C     UPDATE ORBIT AND ERROR MODEL UNKNOWNS
      DO 210 I=1,NUTORB
  210 ORBUNK(I)=ORBUNK(I)+DB(I)
```

```
C     PRINT UPDATED ELEMENTS
      WRITE(6,6004) (ORBUNK(I),I=1,6)
 6004 FORMAT('0   UPDATED ORBIT ELEMENTS IN APPARENT SIDEREAL CARTESIAN
     1 COORDINATES'/5X,'POSITION(METERS)',4X,3F16.3/
     1    5X,'VELOCITY(METERS/SEC)',3F16.6)
      IF(PCODE(15).EQ.0) GO TO 220
C     PUNCH UPDATED ORBIT ELEMENTS.
      WRITE(7,7001) NORB,ORBNAM
 7001 FORMAT(A4,6A8)
      WRITE(7,7002) (ORBUNK(I),I=1,6)
 7002 FORMAT(3F15.3/3F15.6)
  220 CONTINUE
      IF(NEMUNK.EQ.0) GO TO 230
      WRITE(6,6005)
 6005 FORMAT('0  UPDATED VALUE OF ERROR MODEL UNKNOWNS')
      DO 229 I=1,NEMUNK
      MODCOD=MODEL(I,1)
      KSTA=MODEL(I,2)
  229 WRITE(6,6006) (MODALF(J,MODCOD),J=1,3),KORDER(KSTA),(STANAM(J,KSTA
     1),J=1,5),ORBUNK(I+6)
 6006 FORMAT(5X,3A4,'FOR STATION',I6,2X,5A4,2X,'=',F15.3)
  230 CONTINUE
C     UPDATE PASS RECORD ON UNIT 4
      WRITE(4)ORBUNK,NSTATO,LFLG,KSTATO,NEMUNK,EMODEL,MODEL,IGUIDE
     1   , NUTORB
C     RETURN TO PROCESS ANOTHER PASS
      GO TO 150
C
C     ENTER HERE WHEN ALL PASSES HAVE BEEN PROCESSED
  700 CONTINUE
      REWIND 1
      REWIND 4
      IF(PCODE(14).EQ.0) GO TO 730
C     UPDATE COORDINATES OF CENTER OF MASS
      DO 710 I=1,3
  710 STAUVW(I,NSTA)=STAUVW(I,NSTA)+DX(I,NSTA)
      WRITE(6,6007) (STAUVW(I,NSTA),I=1,3),COVX
 6007 FORMAT(////5X,'UPDATED COORDINATES OF THE CENTER OF MASS'/
     1   5X,3F15.3//5X,'WEIGHT COEFFICIENT MATRIX'/3(5X,3F15.3/))
C     RESET NSTA TO ACTUAL NUMBER OF GROUND OBSERVING STATIONS
      NSTA=NSTA-1
  730 CONTINUE
      RETURN
      END
```

```
      FUNCTIONMJD(DATE,MONTH,YEAR)
C     COMPUTATION OF MODIFIED JULIAN DAY
      INTEGER*2 DATE,YEAR
      DIMENSIONMONTHS(2,12)
      DATAMONTHS/3HJAN,0,3HFEB,31,3HMAR,59,3HAPR,90,3HMAY,120,3HJUN,151,
     13HJUL,181,3HAUG,212,3HSEP,243,3HOCT,273,3HNOV,304,3HDEC,334/
      ID=365*(YEAR-50)+(YEAR-49)/4
      DO20I=1,12
      IF(MONTH.EQ.MONTHS(1,I))GOTO25
   20 CONTINUE
      IF(MONTH.LE.12) GO TO 21
      WRITE(6,6001) MONTH
 6001 FORMAT(3X,22HMONTH NAME MISPELLED   ,A3)
      STOP
   21 I=MONTH
      MONTH=MONTHS(1,I)
   25 CONTINUE
      ID=ID+MONTHS(2,I)
      IF(MOD(YEAR*1,4).EQ.0.AND.I.GT.2) ID=ID+1
      MJD=ID+DATE+33281
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION GSTD(T)
COMPUTATION OF GREENWICH SIDEREAL TIME
      DOUBLEPRECISIONT,DGST
      REAL*8 PI2/6.28318530717958/
      DGST=0.277987616D0+1.002737811910*(T-0.33282D5)
C   LINEAR TERM COEFFICIENT SHOULD BE 1.002737811906  BY A.E. SUPP.
      DGST=DGST-DBLE(FLOAT(IDINT(DGST)))
      GSTD=DGST*PI2
      RETURN
      END
```

```
      SUBROUTINEPRENUT(DT,PAN)
COMPUTATION OF PRECESSION AND NUTATION SINCE 1950.0
C     THE DIFFERENCE BETWEEN MODIFIED AND TRUE SIDEREAL
C     TIME
C     DT IS MODIFIED JULIAN DATE OF EPOCH
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION FUNARG(5)
      REAL*4 COEFF(5,13)              /4*0.0,1.0,4*0.0,2.0,2*0.0,2.0,-2.0
     1,2.0,0.0,1.0,4*0.0,1.0,2.0,-2.0,2.0,0.0,-1.0,2.0,-2.0,2.0,2*0.0,2.
     20,-2.0,1.0,2*0.0,2.0,0.0,2.0,1.0,6*0.0,2.0,0.0,2*1.0,0.0,2.0,0.0,2
     3.0,1.0,2*0.0,-2.0,0.0,-1.0,0.0,2.0,0.0,2.0/
      REAL*4 TCOEF(2,13)              /-172327.0,-173.7,2088.0,0.2,-12729
     1.0,-1.3,1261.0,-3.1,-497.0,1.2,214.0,-0.5,124.0,0.1,-2037.0,-0.2,6
     275.0,0.1,-342.0,-0.4,-261.0,0.0,-149.0,0.0,114.0,0.0/
      REAL*8 FUNCOF(3,5)              /.82251280093,.36291645684716D-1,19
     113865.0D-20,.99576620370,.27377785192790-2,-31233.0D-20,.031252469
     214,.036748195691688,-668609.0D-20,.97427079475,.033863192198393,-2
     399023.0D-20,.71995354167,-0.14709422833320-3,432630.0D-20/
      DATARPS/4.848136BD-6/,TPI/6.28318530717958/
      COSE=0.91739033
      NTERMS=13
      BT=DT-15019.5
      BT2=BT*BT
      TT=BT/36525.0
      DO50I=1,5
      FARG=FUNCOF(1,I)+FUNCOF(2,I)*BT+FUNCOF(3,I)*BT2
   50 FUNARG(I)=(FARG-DBLE(FLOAT(IDINT(FARG))))*TPI
C
      DLONG=0.0
      DO80I=1,NTERMS
      ARG=0.0
      DO60J=1,5
   60 ARG=ARG+COEFF(J,I)*FUNARG(J)
      TERM=(TCOEF(1,I)+TCOEF(2,I)*TT)*DSIN(ARG)
      DLONG=DLONG+TERM
   80 CONTINUE
      DMU=DLONG*COSE*0.0001
C    COMPUTE PRECESSION SINCE 1950.0 # KAPPA & OMEGA
      BY=(DT-0.33281923D5)/365.2422D0
      PRECES=(46.0990+1.39E-4*BY)*BY
      PAN=(PRECES+DMU)*RPS
      RETURN
      END
```

```
      SUBROUTINEUVWD(A,B,PHI,LAMDA,H,U,V,W)
C     DOUBLE PRECISION VERSION OF UVW JAN 5, 1968
      DOUBLEPRECISIONPHI,LAMDA,N,E2,FAC,U,V,W,SP
      REAL*8 A,B,H
      E2=1.0-(B/A)**2
      SP=DSIN(PHI)
      N=A/DSQRT(1.0-E2*SP*SP)
      FAC=(N+H)*DCOS(PHI)
      U=FAC*DCOS(LAMDA)
      V=FAC*DSIN(LAMDA)
      W=(N*(1.0-E2)+H)*SP
      RETURN
      END
```

```
      SUBROUTINE DEDIT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/DEDITC/ALFS(50),DEC(50),U(3,50),S(3),D(50),SDC(3,50),SUM,
     1GAST,STAXYZ(3,50),GQI,
     2TD,KSTATE(50),IPASS(50),NSTE,NSUSED,KODE
C  EDIT DATA BASED ON PRELIMINARY STATION POSITIONS AND DELETE BAD
C      OBSERVATIONS AND BAD EVENTS,BASED ON THE DISTANCE CRITERION TD
C  THIS SUBROUTINE IS DIMENSION FOR A MAXIMUM OF MAXSTE=50 STATIONS
C      PARTICIPATING IN ANY ONE EVENT. ALL AFFECTED ARRAYS ARE IN
C      COMMON BLOCK /DEDITC/.
C
C  THE NUMBER OF STATIONS PARTICIPATING IN THE EVENT IS NSTE.
C  THE NUMBER OF STATIONS NOT DELETED IS NSUSED.
C
      COMMON/STALOC/STAUVW(3,150)
      DIMENSION Q(3,3),RHS(3),QI(3,3),VI(3)
C
      MAXSTE=50
C  INITIALIZE
      KODE=1
      DO 110 IS=1,NSTE
  110 IPASS(IS)=1
C          IPASS=1 MEANS THIS DIRECTION OK
C          IPASS=2 MEANS THIS DIRECTION DELETED FROM EVENT
C
C  FORM UNIT VECTORS FOR ALL DIRECTIONS IN THIS EVENT
      DO 125 IS=1,NSTE
      STS=ALFS(IS)-GAST
      CA=DCOS(STS)
      SA=DSIN(STS)
      CD=DCOS(DEC(IS))
      SD=DSIN(DEC(IS))
      U(1,IS)=CA*CD
      U(2,IS)=SA*CD
      U(3,IS)=SD
  125 CONTINUE
C
C  INITIALIZE ARRAYS FOR THIS ITERATION
  130 CONTINUE
      NSUSED=0
      DO 140 I=1,3
      RHS(I)=0.0
      S(I)=0.0
      DO 140 J=1,3
      Q(I,J)=0.0
  140 CONTINUE
C
C  ACCUMULATE EQUATIONS
      DO 190 IS=1,NSTE
      IF(IPASS(IS).EQ.2) GO TO 190
      NSUSED=NSUSED+1
      DO 170 I=1,3
      DO 169 J=1,3
  169 QI(I,J)=U(I,IS)*U(J,IS)
  170 QI(I,I)=QI(I,I)-1.0
      DO 175 I=1,3
      DO 175 J=1,3
      Q(I,J)=Q(I,J)+QI(I,J)
```

```
      RHS(I)=RHS(I)+QI(I,J)*STAXYZ(J,IS)
  175 CONTINUE
  190 CONTINUE
C
C TEST FOR DELETION OF WHOLE EVENT
      IF(NSUSED.LT.2) GO TO 420
C
C  INVERT AND SOLVE
C     THE SATELLITE POSITION S IS SELECTED IN SUCH A WAY THAT THE SUM OF
C     THE SQUARES OF THE DISTANCES FROM S OF THE NON-DELETED RAYS IS MINIMIZED.
      DET=1.0
      CALLDMINV(Q,3,DET,QI(1,1),QI(1,2))
      GQI=DABS(DET/DFLOAT(NSUSED))
      IF(GQI.LT.1.0D-4) GO TO 430
      CALL DGMPRD(Q,RHS,S,3,3,1)
C
C  COMPUTE DISTANCE FROM S FOR EACH RAY
      ISMAX=0
      DMAX=0.0
      SUM=0.0
      DO 280 IS=1,NSTE
      DO 270 I=1,3
      DO 269 J=1,3
  269 QI(I,J)=U(I,IS)*U(J,IS)
      QI(I,I)=QI(I,I)-1.0
      VI(I)=S(I)-STAXYZ(I,IS)
  270 CONTINUE
      DDI=DPDOT(VI,U(1,IS),3)
      DDI=DABS(DDI)
      DI=0.0
      DO 275 I=1,3
      DI=DI+(VI(I)-DDI*U(I,IS))**2
      SDC(I,IS)=VI(I)
  275 CONTINUE
      D(IS)=DSQRT(DI)/DDI*206264.80625
      IF(IPASS(IS).EQ.2) GO TO 280
      SUM=SUM+DI
C  TEST D AGAINST TD AND DELETE IF NECESSARY
      IF(D(IS).LT.DMAX) GO TO 280
      DMAX=D(IS)
      ISMAX=IS
  280 CONTINUE
      IF(DMAX.LT.TD) RETURN
      IPASS(ISMAX)=2
C
C   GO BACK AND MAKE ANOTHER PASS THROUGH THE DATA
      GO TO 130
  400 CONTINUE
C  DELETE WHOLE EVENT
      DO 410 IS=1,NSTE
  410 IPASS(IS)=2
      NSUSED=0
      RET RN
  420 CONTINUE
C  DELETE FOR INSUFFICIENT GOOD OBSERVATIONS
      KODE=2
      GO TO 400
C   DELETE FOR INSUFFICIENT GEOMETRICAL SEPARATION BETWEEN OBSERVATIONS
```

```
      SUBROUTINE RODATA
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      INTEGER*4 ENDSIG/1HE/,CONTIN,DELCOD(2)/1H ,1H*/,ECODE
      INTEGER*2 PLUS/1H+/
      INTEGER*2 ISGNP,IPHID,IPHIM,LONGD,LONGM,ISGNL
      INTEGER*2 ID(50),KEY(50),IHR(50),MIN(50),IDAY(50),IYR(50),IRAH(50)
     1,IRAM(50),ISGND(50),IDECD(50),IDECM(50),IDAT(50,11)
      COMMON/DEDITC/ALFS(50),DEC(50),U(3,50),S(3),D(50),SDC(3,50),EVSUM,
     1GAST,STAXYZ(3,50),GQI,
     2TD,KSTATE(50),IPASS(50),NSTE,NSUSED,ECODE
      COMMON/NSTA/NSTA
      COMMON/STAORD/KORDER(150)
      INTEGER STANAM,IDS*2
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      DIMENSION PM(3,3),AP(2,3)
      DIMENSION MONTH(50)
      EQUIVALENCE(ID(1),IDAT(1,1)),(KEY(1),IDAT(1,2)),(IHR(1),IDAT(1,3))
     1,(MIN(1),IDAT(1,4)),(IDAY(1),IDAT(1,5)),(IYR(1),IDAT(1,6)),(IRAH(1
     2),IDAT(1,7)),(IRAM(1),IDAT(1,8)),(ISGND(1),IDAT(1,9)),(IDECD(1),ID
     3AT(1,10)),(IDECM(1),IDAT(1,11))
      DIMENSION SEC(50),RAS(50),DECS(50),VARRA(50),VARDEC(50),COVRAD(50)
     1,DAT(50,6)
      EQUIVALENCE(SEC(1),DAT(1,1)),(RAS(1),DAT(1,2)),(DECS(1),DAT(1,3)),
     1(VARRA(1),DAT(1,4)),(VARDEC(1),DAT(1,5)),(COVRAD(1),DAT(1,6))
      COMMON/OBSD/OBSD(150),OVOBSD
C
      IF(PCODE(1).EQ.1) GO TO 3
      IF(PCODE(1).EQ.7) GO TO 3
      RETURN
C
    3 MAXSTE=50
      PI=3.14159265358D0
      SPR=206264.80625D0
      PI2=2.0*PI
      WPWSP=0.0
C
      READ(5,5004) TD,OVOBSD
      WRITE(6,6004) TD
 5004 FORMAT(F20.2,F10.2)
 6004 FORMAT(//20X,'TEST DISTANCE =',F20.2,'     SECONDS OF ARC')
      WRITE(3) TD
C   START DATA INPUT
      IEVENT=0
      KEVENT=0
      EPR=0.0
      IS=0
C
C
C  ENTER HERE FOR A NEW OBSERVATION
C
  200 IS=IS+1
      IF(PCODE(1).EQ.7) GO TO 211
C     ENTER HERE IF THE OPTICAL DATA IS IN THE KRAKIWSKY FORMAT
  205 CONTINUE
      READ(5,1021,END=901) ID(IS),KEY(IS),IHR(IS),MIN(IS),SEC(IS),
```

```
     XIDAY(IS),
     1MONTH(IS),IYR(IS),IRAH(IS),IRAM(IS),RAS(IS),ISGND(IS),IDECD(IS),
     2IDECM(IS),DECS(IS),VARRA(IS),VARDEC(IS),COVRAD(IS),CONTIN
 1021 FORMAT(I3,A1,1X,I2,I3,F9.4,I2,A3,I2,2I3,F9.4,A1,I2,I3,F8.4,2X,
     13F5.2,7X,A1)
 1022 FORMAT(14X,I4,5I2,F6.4,1X,2I2,F5.3,A1,2I2,F4.2,26X,A1)
      IF(CONTIN.EQ.ENDSIG) GO TO 250
      DDT=DFLOAT(MJD(IDAY(IS),MONTH(IS),IYR(IS)))
      DDT=DDT+(DFLOAT((IHR(IS)*60+MIN(IS))*60)+SEC(IS))/864.0D2
      IF(IS.LE.1) GO TO 210
C     THIS TEST SHOULD BE TRUE ONLY FOR THE FIRST CARD OF THE FIRST EVENT.
C
CHECH FOR END OF EVENT, ALLOWING 0.5 MS DISCREPANCY
      IF(DABS(DDT-EPR).GT.0.58D-8) GO TO 250
C
C   ENTER HERE TO BEGIN A NEW EVENT
C        THE FIRST ENTRY OF THE EVENT SHOULD ALWAYS BE MADE WITH IS=1
  210 CONTINUE
      IDD=ID(IS)
      KSTA=KSTAID(IDD)
      IF(KSTA.GT.0) GO TO 220
      WRITE(6,6042) ID(IS),IHR(IS),MIN(IS),SEC(IS),IDAY(IS),MONTH(IS),
     1IYR(IS)
 6042 FORMAT(5X, 'STATION NUMBER NOT FOUND IN INPUT LIST',I5,3X,2I3,
     1F8.4,3X,I3,A3,I2,'OBSERVATION IGNORED')
      IF(PCODE(1).EQ.1) GO TO 205
C     ENTER HERE IF THE OPTICAL DATA IS IN THE GEOS FORMAT
  211 CONTINUE
      READ(5,5000,END=901)ID(IS),IYR(IS),MONTH(IS),IDAY(IS),IHR(IS),
     1MIN(IS),SEC(IS),IRAH(IS),IRAM(IS),RAS(IS),ISGND(IS),IDECD(IS),
     2IDECM(IS),DECS(IS),CONTIN,VARRA(IS),VARDEC(IS),COVRAD(IS)
 5000 FORMAT(14X,I4,5I2,F6.4,I3,I2,F5.3,A1,2I2,F4.2,17X,A1,2F3.2,F3.1)
      IF(CONTIN.EQ.ENDSIG) GO TO 250
      DDT=DFLOAT(MJD(IDAY(IS),MONTH(IS),IYR(IS)))
      DDT=DDT+(DFLOAT((IHR(IS)*60+MIN(IS))*60)+SEC(IS))/864.0D2
      IF(IS.LE.1) GO TO 212
C     THIS TEST SHOULD BE TRUE ONLY FOR THE FIRST CARD OF THE FIRST EVENT.
C
CHECH FOR END OF EVENT, ALLOWING 0.5 MS DISCREPANCY
      IF(DABS(DDT-EPR).GT.0.58D-8) GO TO 250
C
C   ENTER HERE TO BEGIN A NEW EVENT
C        THE FIRST ENTRY OF THE EVENT SHOULD ALWAYS BE MADE WITH IS=1
  212 CONTINUE
      IDD=ID(IS)
      KSTA=KSTAID(IDD)
      IF(KSTA.GT.0) GO TO 220
      WRITE(6,6042) ID(IS),IHR(IS),MIN(IS),SEC(IS),IDAY(IS),MONTH(IS),
     1IYR(IS)
      GO TO 211
  220 CONTINUE
  219 IF(PCODE(12).EQ.1) GO TO 221
      IF(PCODE(12).EQ.2) GO TO 222
      GO TO 230
  221 VARRA(IS)=OBSD(KSTA)
      VARDEC(IS)=OBSD(KSTA)
      COVRAD(IS)=0.0
      GO TO 230
```

```
  222 VARRA (IS)=OVOBSD
      VARDEC(IS)=OVOBSD
      COVRAD(IS)=0.0
      GO TO 230
  230 CONTINUE
      KSTATE(IS)=KSTA
      EPR=DDT
      GO TO 200
C
C END OF INPUT FOR THIS EVENT. BEGIN PROCESSING
  250 CONTINUE
      NSTE=IS-1
      IEVENT=IEVENT+1
      GAST=GSTD(EPR)
      CALL PRENUT(EPR,PAN)
      GAST=GAST+PAN
      CALL POLE(EPR,XP,YP)
      XP=XP/SPR
      YP=YP/SPR
      XP=0.0D0
      YP=0.0D0
COMPUTE STATION POSITION IN INSTATANEOUS TERRESTRIAL SYSTEM
      PM(1,1)=1.
      PM(1,2)=0.0
      PM(1,3)=-XP
      PM(2,1)=0.0
      PM(2,2)=1.0
      PM(2,3)=YP
      PM(3,1)=XP
      PM(3,2)=-YP
      PM(3,3)=1.0
C
      DO 270 IS=1,NSTE
      ISGNL=PLUS
      RA=ANRADD(ISGNL,IRAH(IS),IRAM(IS),RAS(IS))*15.0
      ALFS(IS)=RA
      DEC(IS)=ANRADD(ISGND(IS),IDECD(IS),IDECM(IS),DECS(IS))
  270 CONTINUE
      WRITE(3) IEVENT,NSTE,GAST,PM,EPR,
     1((IDAT(IS,J),J=1,11),MONTH(IS),(DAT(IS,J),J=1,6),ALFS(IS),DEC(IS),
     2KSTATE(IS),IS=1,NSTE),CONTIN
C    TEST FOR END OF INPUT
      IF(CONTIN.EQ.ENDSIG) GO TO 700
C  PREPARE FOR NEXT EVENT
      DO 610 I=1,6
  610 DAT(1,I)=DAT(NSTE+1,I)
      MONTH(1)=MONTH(NSTE+1)
      DO 611 I=1,11
  611 IDAT(1,I)=IDAT(NSTE+1,I)
C  RETURN TO START A NEW EVENT
      IS=1
      GO TO 210
C
  700 RETURN
C
C   ERROR EXITS
  901 CONTINUE
C   ENTER HERE IF END SIGNAL CARD IS MISSING FROM INPUT DATA DECK
```

```
      SUBROUTINE RCONAP
C
C
C            CONSTRAINT CODE DIRECTORY
C
C                WEIGHTED CONSTRAINTS
C
C  1  CONSTRAIN  THE COORDINATES OF A STATION AT A PRIORI VALUES*(I.E.WEIGHT IT)
C  2  IMPOSE CHORD DISTANCE CONSTRAINT*.
C  3  IMPOSE RELATIVE POSITION CONSTRAINT*
C  4  IMPOSE DIRECTION CONSTRAINT*
C  5  CONSTRAIN THE GEODETIC LATITUDE,LONGITUDE AND HEIGHT OF A STATION.*
C
C               ABSOLUTE CONSTRAINTS
C
C 11  DEFINE THE ORIGIN OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C 12  DEFINE THE ORIENTATION OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT
C 13  DEFINE THE SCALE OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C 14  COMPLETELY FIX ONE OR MORE COORDINATES OF A STATION * $
C 15  COMPLETELY FIX ONE OR MORE COORDINATES  OF RELATIVE POSITION*$
C
C    *IF THE COORDINATES,RELATIVE POSITION, DISTANCE, OR DIRECTION, TO BE
C     CONSTRAINED ARE NOT GIVEN, THE CONSTRAINT IS COMPUTED FROM THE
C     APPROXIMATE COORDINATES OF THE STATION(S) INVOLVED
C
C     $THE DIAGONAL ELEMENTS OF THE WIEGHT MATRIX ARE
C      WHICH COORDINATES ARE TO BE FIXED. A NON-ZERO CODE MEANS TO FIX
C      THE COORDINATE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER ENDSIG/1HE/,CONTIN,STANAM
      INTEGER*2 IDS
      DIMENSION XI(3),XJ(3),W(3,3),DIS(3),DXB(3),DXC(3)
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/NSTA/NSTA/STAORD/KORDER(150)
      DATA DPR/57.2957795130D0/
   10 READ(5,5000,END=1000)KODE,CONTIN
 5000 FORMAT(I2,77X,A1)
      WRITE(3) KODE,CONTIN
      IF(CONTIN.EQ.ENDSIG) GO TO 1000
      IF(KODE.LE.0) GO TO 950
      IF(KODE.GT.19) GO TO 950
      GO TO (100,200,300,400,500,600,700,800,900,950,
     11100,1200,1300,1400,1500,1600,1700,1800,1900),KODE
C
  100 CONTINUE
      READ(5,5001) IS
 5001 FORMAT(14I5)
      READ(5,5002) XI
 5002 FORMAT(3D16.8)
      READ(5,5002) (W(I,I),I=1,3)
      ISTA=KSID2(IS)
      DO 110 I=1,3
      IF(XI(I).EQ.0.0) XI(I)=STAUVW(I,ISTA)
  110 CONTINUE
      WRITE(3) IS,ISTA,XI,W
      GO TO 10
```

```
C                              CHORD CONSTRAINT
  200 CONTINUE
      READ(5,5001) IS,JS
      READ(5,5002) CD,RELUNC
      ISTA=KSID2(IS)
      JSTA=KSID2(JS)
      CDC=0.0
      DO 205 I=1,3
      DIS(I)=STAUVW(I,ISTA)-STAUVW(I,JSTA)
  205 CDC=CDC+DIS(I)**2
      CDC=DSQRT(CDC)
      IF(CD.EQ.0.0)CD=CDC
      WRITE(3) IS,ISTA,JS,JSTA,CD,RELUNC
      GO TO 10
C                         RELATIVE POSITION CONSTRAINT
  300 CONTINUE
      READ(5,5001) IS,JS
      READ(5,5002) DXB
      READ(5,5002) (W(I,I),I=1,3)
      ISTA=KSID2(IS)
      JSTA=KSID2(JS)
      DO 310 I=1,3
      DXC(I)=STAUVW(I,ISTA)-STAUVW(I,JSTA)
      IF(DXB(I).EQ.0.0) DXB(I)=DXC(I)
  310 CONTINUE
      WRITE(3) IS,ISTA,JS,JSTA,DXB,W
      GO TO 10
C
  400 CONTINUE
C
C                DIRECTION CONSTRAINTS
C                ALPHA IS LONGITUDE-LIKE ANGLE
C                BETA IS LATITUDE-LIKE ANGLE
      READ(5,5001) IS,JS
      ISTA=KSID2(IS)
      JSTA=KSID2(JS)
C    READ ANGLES IN DEGREES AND UNCERTAINTIES IN SECONDS OF ARC
      READ(5,5002) ALF,BETA
      READ(5,5002) VARA,VARB,COVAB
      DO 405 I=1,3
  405 DXC(I)=STAUVW(I,ISTA)-STAUVW(I,JSTA)
      IF(ALF) 412,411,412
  411 ALF=DATAN2(DXC(2),DXC(1))
      ALF=ALF*DPR
  412 IF(BETA)414,413,414
  413 RSCSB=DXC(1)**2+DXC(2)**2
      BETA=DATAN(DXC(3)/DSQRT(RSCSB))
      BETA=BETA*DPR
  414 CONTINUE
      WRITE(3) IS,ISTA,JS,JSTA,ALF,BETA,VARA,VARB,COVAB
      GO TO 10
C
  500 CONTINUE
C    CONSTRAINT ON GEODETICLATITUDE,LONGITUDE, AND HEIGHT
      READ(5,5001) IS
      ISTA=KSID2(IS)
      IDTS=IDS(ISTA)
C    READ LATITUDE AND LONGITUDE IN DEGREES AND HEIGHT IN METERS
```

```
C    AN INPUT COORDINATE OF ZERO INDICATES THAT THE APPROXIMATE VALUE OF
C         THE COORDINATE IS TO BE USED.
      READ(5,5002) PHIO,FLAMO,HO
C    READ UNCERTAINTIES IN SECONDS OF ARC AND METERS
      READ(5,5002) SDP,SDL,SDH
C    AN INPUT UNCERTAINTY OF ZERO INDICATES THAT A ZERO WEIGHT IS TO BE USED.
      CALL UVWTG2(STAUVW(1,ISTA),DATPRM(1,IDTS),PHI,FLAM,H)
      IF(PHIO.EQ.0.0) PHIO=PHI*DPR
      IF(FLAMO.EQ.0.0) FLAMO=FLAM*DPR
      IF(HO.EQ.0.0) HO=H
      WRITE(3) IS,ISTA,IDTS,PHIO,FLAMO,HO,SDP,SDL,SDH
      GO TO 10
C
  600 CONTINUE
  700 CONTINUE
  800 CONTINUE
  900 CONTINUE
      GO TO 950
C
C     INNER ADJUSTMENT CONSTRAINTS
 1100 CONTINUE
 1200 CONTINUE
 1300 CONTINUE
 1600 CONTINUE
      GO TO 10
C
 1400 CONTINUE
C    FIX A STATION
      GO TO 100
C
 1500 CONTINUE
      GO TO 300
C
 1700 CONTINUE
 1800 CONTINUE
 1900 CONTINUE
      GO TO 950
  950 WRITE(6,6095) KODE
 6095 FORMAT('OILLEGAL CONSTRAINT CODE IN CONAP IGNORED',I5)
      GO TO 10
C
 1000 CONTINUE
      REWIND 3
      RETURN
      END
```

```
C
C           CONSTRAINT CODE DIRECTORY
C
C               WEIGHTED CONSTRAINTS
C
C   1  CONSTRAIN  THE COORDINATES OF A STATION AT A PRIORI VALUES*(I.E.WEIGHT IT)
C   2  IMPOSE CHORD DISTANCE CONSTRAINT*.
C   3  IMPOSE RELATIVE POSITION CONSTRAINT*
C   4  IMPOSE DIRECTION CONSTRAINT*
C   5  CONSTRAIN THE GEODETIC LATITUDE,LONGITUDE AND HEIGHT OF A STATION.*
C
C              ABSOLUTE CONSTRAINTS
C
C  11  DEFINE THE ORIGIN OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C  12  DEFINE THE ORIENTATION OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT
C  13  DEFINE THE SCALE OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C  14  COMPLETELY FIX ONE OR MORE COORDINATES OF A STATION * $
C  15  COMPLETELY FIX ONE OR MORE COORDINATES  OF RELATIVE POSITION*$
C
C     *IF THE COORDINATES,RELATIVE POSITION, DISTANCE, OR DIRECTION, TO BE
C      CONSTRAINED ARE NOT GIVEN, THE CONSTRAINT IS COMPUTED FROM THE
C      APPROXIMATE COORDINATES OF THE STATION(S) INVOLVED
C
C      $THE DIAGONAL ELEMENTS OF THE W MATRIX ARE USED AS CODES TO INDICATE
C       WHICH COORDINATES ARE TO BE FIXED. A NON-ZERO CODE MEANS TO FIX
C       THE COORDINATE.
C
      SUBROUTINE PSOLN
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/NSTA/NSTA/STAORD/KORDER(150)
      INTEGER*2 L,LSOLVE,IDS
      INTEGER STANAM
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/WPW/WPW,XPU,IDEGF,IFSTA
      INTEGER*2 PCODE
      COMMON/PCODES/PCODE(20)
      DIMENSION ADX(3)
      DIMENSION EIG(6),EV(3,3),COVX(3,3),VARX(3,40)
      REAL*8 LAM
      DIMENSION  Q(3,3,40),DELCOV(3,3),UNC(3)
      DIMENSION            DX(3,40),UNCE(3,40)
      DIMENSION RLX(3,3),RLD(3,3),RXD(3,3)
      EQUIVALENCE(RXD(1,1),EV(1,1))
      LOC(K)=(K*(K+1))/2
      REWIND 2
      WRITE(6,6001) WPW,XPU
 6001 FORMAT(/10X,'W''PW=',D16.8,'  -X''U=',D16.8)
      VPV=WPW-XPU
      IDEGF=IDEGF-3*NSTA
      VARO=VPV/DFLOAT(IDEGF)
      SIGO=DSQRT(VARO)
      WRITE(6,6002) IDEGF,VPV,VARO,SIGO
 6002 FORMAT(1H1/////////T50,'NUMBER OF DEGREES OF FREEDOM =',I8/1H0,T38
     1,'QUADRATIC SUM OF ALL THE RESIDUALS (VPV) =',F13.4/1H0,T55,
     2'VARIANCE OF UNIT WEIGHT =',F13.4/1H0,T45,
     3'STANDARD DEVIATION OF UNIT WEIGHT =', F13.4)
C
```

```
      READ(2) ((VARX(I,ISTA),I=1,3),ISTA=1,NSTA)
      DO 80 ISTA=1,NSTA
      DO 80 I=1,3
      IF(VARX(I,ISTA).LE.0.0) GO TO 79
      VARX(I,ISTA)=DSQRT(VARX(I,ISTA))
      GO TO 80
   79 VARX(I,ISTA)=0.0
   80 CONTINUE
C
      DO 200 ISTA=1,NSTA
      READ(2)(DX(I,ISTA),I=1,3),(((Q(I,J,JSTA),I=1,3),J=1,3),
     1JSTA=ISTA,NSTA)
C
      IF(ISTA-2*(ISTA/2).EQ.1.OR.PCODE(20).EQ.1) WRITE(6,6011)
 6011 FORMAT(1H1)
      IDAT=IDS(ISTA)
      WRITE(6,6003) KORDER(ISTA),(STANAM(I,ISTA),I=1,5)
     1,IDAT,(DATNAM(I,IDAT),I=1,4)
 6003 FORMAT(3(/),'OSTATION NUMBER - ',I8,10X,4A4,A2,5X,
     1'ELLIPSOID',I4,4X,4A8)
      WRITE(6,6010)
 6010 FORMAT('0',25X,'X',15X,'Y',15X,'Z',21X,'LAT.',12X,'LONG.(+E)
     1ELL. HT.')
      CALL UVWTG(STAUVW(1,ISTA),DATPRM(1,IDAT),PHI,LAM,H)
      CALL DANG (PHI,ISGNP,IDEGP,IMINP,SECP)
      CALL DANG (LAM,ISGNL,IDEGL,IMINL,SECL)
      WRITE(6,6005) (STAUVW(I,ISTA),I=1,3),ISGNP,IDEGP,IMINP,SECP,
     1ISGNL,IDEGL,IMINL,SECL,H
 6005 FORMAT('OPREL. COORD. - ',3F16.4,7X,2(3X,A1,2I3,F8.4),F12.4)
      DO 100 IK=1,3
  100 ADX(IK)=DX(IK,ISTA)
      CALL DELL(ADX                               ,Q(1,1,ISTA),PHI,LAM,H,
     1 DATPRM(1,IDAT),DP,DL,DH,DELCOV,RLX)
      DO 120 I=1,3
      DO 110 J=1,3
      DELCOV(I,J)=DELCOV(I,J)*VARO
  110 COVX(I,J)=Q(I,J,ISTA)*VARO
      UNC(I)=VARX(I,ISTA)*SIGO
      IF(DELCOV(I,I).GT.0.0) GO TO 115
      UNCE(I,ISTA)=0.0
      GO TO 116
  115 CONTINUE
      UNCE(I,ISTA)=DSQRT(DELCOV(I,I))
  116 CONTINUE
      STAUVW(I,ISTA)=STAUVW(I,ISTA)+DX(I,ISTA)
  120 CONTINUE
C
      CALL DANG(DP,ISGNP,IDEGP,IMINP,SECP)
      CALL DANG(DL,ISGNL,IDEGL,IMINL,SECL)
      DO 125 I=1,3
      IF(I.LT.3) UNCE(I,ISTA)=UNCE(I,ISTA)*206264.8062
      DO 125 J=1,3
      IF(I.LT.3) DELCOV(I,J)=DELCOV(I,J)*206264.8062
      IF(J.LT.3) DELCOV(I,J)=DELCOV(I,J)*206264.8062
  125 CONTINUE
      WRITE(6,6006) (DX(I,ISTA),I=1,3)
     1,ISGNP,IDEGP,IMINP,SECP,ISGNL,IDEGL,IMINL,SECL,DH
 6006 FORMAT('OCORRECTIONS -  ',3F16.4,7X,2(3X,A1,2I3,F8.4),F12.4)
```

```
      CALL UVWTG(STAUVW(1,ISTA),DATPRM(1,IDAT),PHI,LAM,H)
      CALL DANG(PHI,ISGNP,IDEGP,IMINP,SECP)
      CALL DANG(LAM,ISGNL,IDEGL,IMINL,SECL)
      WRITE(6,6007) (STAUVW(I,ISTA),I=1,3)
     1,ISGNP,IDEGP,IMINP,SECP,ISGNL,IDEGL,IMINL,SECL,H
 6007 FORMAT('OADJ. COORD. -   ',3F16.4,7X,2(3X,A1,2I3,F8.4),F12.4)
      IF(PCODE(17).EQ.1) WRITE(7,7001) KORDER(ISTA),(STAUVW(I,ISTA),
     1  I=1,3),(COVX(I,I),I=1,3)
 7001 FORMAT(I4,4X,3F16.6/3F10.3)
      IF(PCODE(18).EQ.1) WRITE(7,5005) KORDER(ISTA),IDAT,
     1(STANAM(I,ISTA),I=1,5),ISGNP,IDEGP,IMINP,SECP,IDEGL,IMINL,SECL,H,
     2(UNCE(I,ISTA),I=1,3)
 5005 FORMAT(I4,I2,4A4,A2,A1,2(2I3,F8.4), F10.2,2F3.1,F3.0,7X,A1)
      WRITE(6,6008) ((COVX(I,J),J=1,3),(DELCOV(I,J),J=1,3),I=1,3)
 6008 FORMAT('OVARIANCE-COVARIANCE MATRIX OF THE STATION POSITION'//
     13(14X,3F16.6, 10X,3F16.6/))
      WRITE(6,6009) UNC,(UNCE(I,ISTA),I=1,3)
 6009 FORMAT('OSTAND. DEV. -',3F16.4,10X,3F16.4)
C
      IF(PCODE(19).NE.1) GO TO 150
C                                        COMPUTE EIGENVALUES
      WRITE(6,6100)
 6100 FORMAT('ODIRECTIONS OF EIGENVECTORS AND SQUARE ROOTS OF EIGENVALUE
     1S OF VARIANCE-COVARIANCE MATRIX -'/T20,'LATITUDE',T40,'LONGITUDE',
     2  T60,'ELEVATION',T80,'AZIMUTH',T100,'AXIS LENGTH')
      NB=0
      DO 135  J=1,3
      DO 135  I=1,J
      NB=NB+1
  135 EIG(NB)=COVX(I,J)
      CALL DEIGEN(EIG,EV,3,0)
      CALL DGMPRD(RLX,RXD,RLD,3,3,3)
      DO 140  I=1,3
      PHI=DATAN(EV(3,I)/DSQRT(EV(1,I)**2+EV(2,I)**2))
      LAM=DATAN2(EV(2,I),EV(1,I))
      ELEV=DATAN (RLD(3,I)/DSQRT(RLD(1,I)**2+RLD(2,I)**2))
      AZ=DATAN2(RLD(2,I),RLD(1,I))
      CALL DANG(PHI,ISGNP,IDEGP,IMINP,SECP)
      CALL DANG(LAM,ISGNL,IDEGL,IMINL,SECL)
      CALL DANG(ELEV,ISGNEL,IDEGEL,IMINEL,SECEL)
      CALL DANG(AZ,ISGNAZ,IDEGAZ,IMINAZ,SECAZ)
      EVAL=EIG(LOC(I))
      IF(EVAL.LE.0.0) GO TO 137
      EVAL=DSQRT(EVAL)
      GO TO 139
  137 EVAL=0.0
  139 CONTINUE
      WRITE(6,6101) ISGNP,IDEGP,IMINP,SECP,ISGNL,IDEGL,IMINL,SECL,
     1  ISGNEL,IDEGEL,IMINEL,SECEL,ISGNAZ,IDEGAZ,IMINAZ,SECAZ,EVAL
 6101 FORMAT(1H0,14X,4(A1,2I3,F8.4,5X),F12.4)
  140 CONTINUE
  150 CONTINUE
C
      IF(PCODE(20).NE.1) GO TO 200
C                                        COMPUTE CORRELATION COEFFICIENTS
      WRITE(6,6105)
 6105 FORMAT(1H0,15X,'3X3 WEIGHT COEFFICIENT MATRICES',43X,'CORRELATION
     1COEFFICIENTS'/1H0,2(25X,'X',15X,'Y',15X,'Z'))
```

```
      DO 160 JSTA=ISTA,NSTA
      DO 155 I=1,3
      DO 155 J=1,3
      DENOM=VARX(I,ISTA)*VARX(J,JSTA)
      IF(DENOM) 154,154,153
  153 EV(I,J)=Q(I,J,JSTA)/DENOM
      GO TO 155
  154 EV(I,J)=0.0
  155 CONTINUE
      WRITE(6,6106) KORDER(ISTA),KORDER(JSTA)
 6106 FORMAT(1H0,5X,' STA. NO.',I5,' WITH STA. NO.',I5)
      WRITE(6,6107) ((Q(I,J,JSTA),J=1,3),(EV(I,J),J=1,3),I=1,3)
 6107 FORMAT(14X,3F16.6,10X,3F16.6)
  160 CONTINUE
  200 CONTINUE
C     PRINT OUT SUMMERY OF RESULTS
      IF(PCODE(10).EQ.0) GO TO 300
      WRITE(6,6011)
      DO 300 ISTA=1,NSTA
      DO 219 I=1,3
  219 UNC(I)=VARX(I,ISTA)*SIGO
      WRITE(6,6108) KORDER(ISTA),(STANAM(I,ISTA),I=1,5)
      IDAT=IDS(ISTA)
      JD=PCODE(10)
      GO TO(220,240,260,260),JD
  220 WRITE(6,6109) (DX(I,ISTA),I=1,3),UNC
      GO TO 300
  240 WRITE(6,6110) (STAUVW(I,ISTA),I=1,3),UNC
      GO TO 300
  260 CALL UVWTG(STAUVW(1,ISTA),DATPRM(1,IDAT),PHI,LAM,H)
      CALL DANG (PHI,ISGNP,IDEGP,IMINP,SECP)
      CALL DANG (LAM,ISGNL,IDEGL,IMINL,SECL)
      IF (PCODE(10).EQ.4) GO TO 270
      WRITE(6,6111) ISGNP,IDEGP,IMINP,SECP,ISGNL,IDEGL,IMINL,SECL,H,
     1(UNCE(I,ISTA),I=1,3)
      GO TO 300
  270 WRITE(6,6112) (STAUVW(I,ISTA),I=1,3),ISGNP,IDEGP,IMINP,SECP,IDEGL,
     1IMINL,SECL,H,UNC,(UNCE(I,ISTA),I=1,3)
 6108 FORMAT(I8,1X,4A4,A2)
 6109 FORMAT (1H+,27X,3F10.4,/28X,3F10.4/)
 6110 FORMAT (1H+,27X,3F16.4,/28X,3F16.4/)
 6111 FORMAT (1H+,27X,2(3X,A1,2I3,F8.4),F12.4,/38X,F8.4,10X,F8.4,F12.4/)
 6112 FORMAT(1H+,27X,3F16.4,  3X,A1,2I3,F8.4,3X,2I3,F8.4,F12.4,/28X,
     1 3F16.4,2(10X,F8.4),F12.4/)
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONAP1(KODE)
C
C           CONSTRAINT CODE DIRECTORY
C
C              WEIGHTED CONSTRAINTS
C
C   1  CONSTRAIN  THE COORDINATES OF A STATION AT A PRIORI VALUES*(I.E.WEIGHT IT)
C   2  IMPOSE CHORD DISTANCE CONSTRAINT*.
C   3  IMPOSE RELATIVE POSITION CONSTRAINT*
C   4  IMPOSE DIRECTION CONSTRAINT*
C   5  CONSTRAIN THE GEODETIC LATITUDE,LONGITUDE AND HEIGHT OF A STATION.*
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 L,LSOLVE,IDS
      INTEGER             CONTIN,STANAM
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/NSTA/NSTA,NBLOCK
      COMMON/STAORD/KORDER(150)
      COMMON/NORMEQ/REDN(3,3,2485),U(3,70),L(2485),LSOLVE
      COMMON/WPW/WPW,XPU,IDEGF,IFSTA
      DIMENSION XI(3),XJ(3),W(3,3),DIS(3),DXB(3),DXC(3)
      EQUIVALENCE (XI(1),DXB(1)),(XJ(1),DXC(1))
      DIMENSION G(2,3)
      DATA SPR,DPR/206264.80625,57.295779513/
      LOC(K)=(K*(K+1))/2
 6150 FORMAT(10X,3D16.8)
      GO TO (100,200,300,400,500,600,700,800,900),KODE
  100 CONTINUE
      READ (3) IS,ISTA,XI,W
      NB=LOC(ISTA)
      K=0
      DO 110 I=1,3
      DIS(I)=XI(I)-STAUVW(I,ISTA)
      REDN(I,I,NB)=REDN(I,I,NB)+W(I,I)
      U(I,ISTA)=U(I,ISTA)+W(I,I)*DIS(I)
      IF(W(I,I).EQ.0.0) GO TO 110
      K=K+1
      WPW=WPW+DIS(I)*W(I,I)*DIS(I)
      IDEGF=IDEGF+1
  110 CONTINUE
      IF(K.EQ.3) L(NB)=L(NB)+1
      WRITE(6,6100) IS,(STANAM(I,ISTA),I=1,5),XI,(W(I,I),I=1,3)
 6100 FORMAT(///15X,'A PRIORI CONSTRAINT ON STATION',I5,2X,5A4/
     115X,'COORDINATES',3F16.2/15X,'WEIGHTS',6X,3F16.4)
      GO TO 10
C                          CHORD CONSTRAINT
  200 CONTINUE
      READ (3) IS,ISTA,JS,JSTA,CD,RELUNC
      IF(JSTA.GE.ISTA) GO TO 210
C  SWITCH SUBSCRIPTS IF NECESSARY
      NB=ISTA
      ISTA=JSTA
      JSTA=NB
  210 CONTINUE
      CDC =0.0
      DO 205 I=1,3
      DIS(I)=STAUVW(I,ISTA)-STAUVW(I,JSTA)
```

```
  205 CDC=CDC+DIS(I)**2
      CDC=DSQRT(CDC)
      DO 215 I=1,3
  215 DIS(I)=DIS(I)/CDC
      CDD=CD-CDC
      WCD=(RELUNC/CD)**2
      IB=LOC(ISTA)
      JB=LOC(JSTA)
      NB=LOC(JSTA-1)+ISTA
      DO 220 I=1,3
      U(I,ISTA)=U(I,ISTA)+DIS(I)*WCD*CDD
      U(I,JSTA)=U(I,JSTA)-DIS(I)*WCD*CDD
      DO 220 J=1,3
      TERM=DIS(I)*WCD*DIS(J)
      REDN(I,J,IB)=REDN(I,J,IB)+TERM
      REDN(I,J,JB)=REDN(I,J,JB)+TERM
      REDN(I,J,NB)=REDN(I,J,NB)-TERM
  220 CONTINUE
      WRITE(6,6200) KORDER(ISTA),(STANAM(I,ISTA),I=1,5),KORDER(JSTA),
     1(STANAM(I,JSTA),I=1,5),CD,RELUNC
 6200 FORMAT(////15X,'CHORD DISTANCE CONSTRAINT IMPOSED BETWEEN STATION'
     1,I5,2X,5A4/53X,'AND STATION',I5,2X,5A4/15X,'CONSTRAINED DISTANCE='
     2,F16.2/15X,'THE WEIGHT IS COMPUTED FROM A RELATIVE UNCERTAINTY OF
     3ONE PART IN',F16.2)
      WRITE(6,6150) CDD
      L(IB)=L(IB)+1
      L(JB)=L(JB)+1
      L(NB)=L(NB)+1
      WPW=WPW+ CDD*WCD*CDD
      IDEGF=IDEGF+1
      GO TO 10
C                       RELATIVE POSITION CONSTRAINT
  300 CONTINUE
      READ (3) IS,ISTA,JS,JSTA,DXB,W
      IB=LOC(ISTA)
      JB=LOC(JSTA)
      NB=LOC(JSTA-1)+ISTA
      IF(ISTA.GT.JSTA) NB=LOC(ISTA-1)+JSTA
      DO 310 I=1,3
      DXC(I)=STAUVW(I,ISTA)-STAUVW(I,JSTA)
      DIS(I)=DXB(I)-DXC(I)
      IF(W(I,I).EQ.0.0) GO TO 310
      WPW=WPW+DIS(I)*W(I,I)*DIS(I)
      IDEGF=IDEGF+1
      U(I,ISTA)=U(I,ISTA)+W(I,I)*DIS(I)
      U(I,JSTA)=U(I,JSTA)-W(I,I)*DIS(I)
      REDN(I,I,IB)=REDN(I,I,IB)+W(I,I)
      REDN(I,I,JB)=REDN(I,I,JB)+W(I,I)
      REDN(I,I,NB)=REDN(I,I,NB)-W(I,I)
  310 CONTINUE
      L(IB)=L(IB)+1
      L(JB)=L(JB)+1
      L(NB)=L(NB)+1
      WRITE(6,6300) KORDER(ISTA),(STANAM(I,ISTA),I=1,5),KORDER(JSTA),
     1(STANAM(I,JSTA),I=1,5),DXB,(W(I,I),I=1,3)
 6300 FORMAT(////15X,'RFLATIVE POSITION CONSTRAINT'/15X,'BETWEEN STATION
     1',I5,3X,5A4,' AND STATION',I5,2X,5A4//15X,'RELATIVE COORDINATES
     2ARE'/15X,3F16.2//15X,'WEIGHTS ARE'/17X,3F16.4)
```

```
      WRITE(6,6150) (DIS(I),I=1,3)
      GO TO 10
  400 CONTINUE
C                DIRECTION CONSTRAINTS
C                ALPHA IS LONGITUDE-LIKE ANGLE
C                BETA IS LATITUDE-LIKE ANGLE
      READ (3) IS,ISTA,JS,JSTA,ALF,BETA,VARA,VARB,COVAB
      WRITE(6,6400) KORDER(ISTA),(STANAM(I,ISTA),I=1,5),KORDER(JSTA),
     1(STANAM(I,JSTA),I=1,5),ALF,BETA,VARA,VARB,COVAB
 6400 FORMAT(////15X,'DIRECTION CONSTRAINT IMPOSED BETWEEN STATION',
     1I5,2X,5A4,/48X,'AND STATION',I5,2X,5A4/15X,'ANGLES(DEGREES)',4X,
     22F16.8/15X,'UNCERTAINTIES(SECONDS)',3F16.3)
      DO 405 I=1,3
  405 DXC(I)=STAUVW(I,ISTA)-STAUVW(I,JSTA)
      RSCSB=DXC(1)**2+DXC(2)**2
      TA=DXC(2)/DXC(1)
      CSA=1.0/(1.0+TA*TA)
      RCB=DSQRT(RSCSB)
      TB=DXC(3)/RCB
      CSB=1.0/(1.0+TB*TB)
C
      AO=DATAN2(DXC(2),DXC(1))
      DIS(1)=ALF/DPR-AO
      PI=180.0/DPR
      IF(DIS(1).GT.PI) DIS(1)=DIS(1)-2.0*PI
      IF(DIS(1).LT.(-PI)) DIS(1)=DIS(1)+2.0*PI
      BO=DATAN(DXC(3)/RCB)
      DIS(2)=BETA/DPR-BO
      WRITE(6,6150) DIS(1),DIS(2)
C
      G(1,1)=CSA*TA/DXC(1)
      G(1,2)=-CSA/DXC(1)
      G(1,3)=0.0
      G(2,1)=CSB*TB*DXC(1)/RSCSB
      G(2,2)=G(2,1)*TA
      G(2,3)=-CSB/RCB
C
      VARA=(VARA/SPR)**2
      VARB=(VARB/SPR)**2
      DET=VARA*VARB-COVAB*COVAB
      W(1,1)=VARB/DET
      W(2,2)=VARA/DET
      W(1,2)=-COVAB/DET
      W(2,1)=W(1,2)
C
      IB=LOC(ISTA)
      JB=LOC(JSTA)
      NB=LOC(JSTA-1)+ISTA
      IF(ISTA.GT.JSTA) NB=LOC(ISTA-1)+JSTA
      DO 445 I=1,3
      SUM=0.0
      DO 443 II=1,2
      DO 443 JJ=1,2
  443 SUM=SUM+G(II,I)*W(II,JJ)*DIS(JJ)
      U(I,ISTA)=U(I,ISTA)-SUM
      U(I,JSTA)=U(I,JSTA)+SUM
      DO 445 J=1,3
      SUM=0.0
```

```
      DO 444   II=1,2
      DO 444   JJ=1,2
  444 SUM=SUM+G(II,I)*W(II,JJ)*G(JJ,J)
      REDN(I,J,IB)=REDN(I,J,IB)+SUM
      REDN(I,J,JB)=REDN(I,J,JB)+SUM
      IF(ISTA.GT.JSTA) GO TO 446
      REDN(I,J,NB)=REDN(I,J,NB)-SUM
      GO TO 445
  446 REDN(J,I,NB)=REDN(J,I,NB)-SUM
  445 CONTINUE
      DO 450 II=1,2
      IF(W(II,II).EQ.0.0) GO TO 450
      L(IB)=L(IB)+1
      L(JB)=L(JB)+1
      L(NB)=L(NB)+1
      IDEGF=IDEGF+1
      DO 450 JJ=1,2
      WPW=WPW+DIS(II)*W(II,JJ)*DIS(JJ)
  450 CONTINUE
      GO TO 10
C
  500 CONTINUE
      READ (3) IS,ISTA,IDTS,PHIO,FLAMO,HO,SDP,SDL,SDH
      WRITE(6,6500) IS,(STANAM(I,ISTA),I=1,5),PHIO,FLAMO,HO,IDTS,
     1 (DATNAM(I,IDTS),I=1,4), SDP,SDL,SDH
 6500 FORMAT(////15X,'THE ELLIPSOIDAL COORDINATES (LAT.,LONG.,HEIGHT) OF
     1 STATION',I7,3X,5A4/15X,'ARE CONSTRAINED AT'//2(F20.9,' DEGREES'),
     2  F20.3,' METERS'//15X,'ON DATUM',I5,3X,4A8//
     3 15X,'THE WEIGHTS FOR THESE CONSTRAINTS ARE COMPUTED FROM OBSERVAT
     4IONAL STANDARD DEVIATIONS OF'//
     5 2(F20.3,' SECONDS'),F20.3,' METERS')
      CALL UVWTG3(STAUVW(1,ISTA),DATPRM(1,IDTS),PHI,FLAM,H)
      IB=LOC(ISTA)
      SP=DSIN(PHI)
      CP=DCOS(PHI)
      SL=DSIN(FLAM)
      CL=DCOS(FLAM)
      AE=DATPRM(1,IDTS)
      E2=1.0-(DATPRM(2,IDTS)/AE)**2
      EW=DSQRT(1.0-E2*SP*SP)
      EN=AE/EW
      EM=AE*(1.0-E2)/EW**3
      IF(SDP.EQ.0.0) GO TO 510
      WT=1.0/(SDP/SPR)**2
      DXB(1)=-SP*CL/(EM+H)
      DXB(2)=-SP*SL/(EM+H)
      DXB(3)=     CP/(EM+H)
      DISC=PHIO/DPR-PHI
      ASSIGN 510 TO J5
      GO TO 550
  510 CONTINUE
      IF(SDL.EQ.0.0) GO TO 520
      WT=1.0/(SDL/SPR)**2
      DENOM=(EN+H)*CP
      DXB(1)=-SL/DENOM
      DXB(2)= CL/DENOM
      DXB(3)=0.0
      DISC=FLAMO/DPR-FLAM
```

```
      ASSIGN 520 TO J5
      GO TO 550
  520 CONTINUE
      IF(SDH.EQ.0.0) GO TO 560
      WT=1.0/SDH**2
      DXB(1)=CP*CL
      DXB(2)=CP*SL
      DXB(3)=SP
      DISC=HO-H
      WRITE(6,6150) DISC
      ASSIGN 560 TO J5
      GO TO 550
  550 CONTINUE
      DO 555 I=1,3
      U(I,ISTA)=U(I,ISTA)+DXB(I)*WT*DISC
      DO 555 J=1,3
      REDN(I,J,IB)=REDN(I,J,IB)+DXB(I)*WT*DXB(J)
  555 CONTINUE
      L(IB)=L(IB)+1
      IDEGF=IDEGF+1
      WPW=WPW+DISC*WT*DISC
      GO TO J5,(510,520,560)
  560 CONTINUE
      GO TO 10
C
  600 CONTINUE
  700 CONTINUE
  800 CONTINUE
  900 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONAP2(KODE2)
C             ABSOLUTE CONSTRAINTS
C
C 11  DEFINE THE ORIGIN OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C 12  DEFINE THE ORIENTATION OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT
C 13  DEFINE THE SCALE OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
C 14  COMPLETELY FIX ONE OR MORE COORDINATES OF A STATION *
C 15  COMPLETELY FIX ONE OR MORE COORDINATES  OF RELATIVE POSITION*$
C
C    *IF THE COORDINATES,RELATIVE POSITION, DISTANCE, OR DIRECTION, TO BE
C     CONSTRAINED ARE NOT GIVEN, THE CONSTRAINT IS COMPUTED FROM THE
C     APPROXIMATE COORDINATES OF THE STATION(S) INVOLVED
C
C     $THE DIAGONAL ELEMENTS OF THE W MATRIX ARE USED AS CODES TO INDICATE
C      WHICH COORDINATES ARE TO BE FIXED. A NON-ZERO CODE MEANS TO FIX
C      THE COORDINATE.
C
C  PROCESS A PRIORI CONSTRAINTS ON, AND BETWEEN, STATIONS
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 L,LSOLVE,IDS
      INTEGER ENDSIG/1HE/,CONTIN,STANAM
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/NSTA/NSTA,NBLOCK
      COMMON/STAORD/KORDER(150)
      COMMON/NORMEQ/REDN(3,3,2485),U(3,70),L(2485),LSOLVE
      COMMON/WPW/WPW,XPU,IDEGF,IFSTA
      DIMENSION XI(3),XJ(3),W(3,3),DIS(3),DXB(3),DXC(3)
      EQUIVALENCE (XI,DXB),(XJ,DXC)
      DIMENSION G(2,3)
      DATA SPR/206264.80625/
      LOC(K)=(K*(K+1))/2
      MAXBLK=70
C
C   ABSOLUTE CONSTRAINTS THAT REQUIRE EXPANSION OF THE NORMAL EQUATION
C         MATRIX BY THE ADDITION OF LAGRANGE MULTIPLIERS
C
C   KODE2 IS KODE-10
      GO TO (1100,1200,1300,1400,1500,1600,1700,1800,1900),KODE2
C 11  DEFINE THE ORIGIN OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
 1100 CONTINUE
      ASSIGN 1110 TO JRTN
      GO TO 960
 1110 CONTINUE
      DO 1120 ISTA=1,NSTA
      IB=LOC(NBLOCK-1)+ISTA
      L(IB)=1
      DO 1120 I=1,3
 1120 REDN(I,I,IB)=1.0
      IDEGF=IDEGF+3
      WRITE(6,6011)
 6011 FORMAT('OTHE ORIGIN OF THE COORDINATE SYSTEM IS DEFINED BY INNER
     1ADJUSTMENT.')
      GO TO 10
C 12  DEFINE THE ORIENTATION OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT
 1200 CONTINUE
      ASSIGN 1210 TO JRTN
      GO TO 960
```

```
 1210 CONTINUE
      DO 1220 ISTA=1,NSTA
      IB=LOC(NBLOCK-1)+ISTA
      REDN(1,2,IB)= STAUVW(3,ISTA)/SPR
      REDN(1,3,IB)=-STAUVW(2,ISTA)/SPR
      REDN(2,1,IB)=-STAUVW(3,ISTA)/SPR
      REDN(2,3,IB)= STAUVW(1,ISTA)/SPR
      REDN(3,1,IB)= STAUVW(2,ISTA)/SPR
      REDN(3,2,IB)=-STAUVW(1,ISTA)/SPR
      L(IB)=1
 1220 CONTINUE
      IDEGF=IDEGF+3
      WRITE(6,6012)
 6012 FORMAT('OORIENTATION OF THE COORDINATE SYSTEM DEFINED BY INNER
     1ADJUSTMENT PROCEDURE')
      GO TO 10
C 13  DEFINE THE SCALE OF THE COORDINATE SYSTEM BY INNER ADJUSTMENT EQUATIONS
 1300 CONTINUE
      ASSIGN 1310 TO JRTN
      GO TO 960
 1310 CONTINUE
      DO 1320 ISTA=1,NSTA
      IB=LOC(NBLOCK-1)+ISTA
      L(IB)=1
      DO 1320 I=1,3
      REDN(I,1,IB)=STAUVW(I,ISTA)/1000000.0
 1320 CONTINUE
C  FILL IN EXTRA TWO ROWS IN BLOCK WITH DUMMY EQUATIONS
      NB=LOC(NBLOCK)
      REDN(2,2,NB)=1.0
      REDN(3,3,NB)=1.0
      L(NB)=1
      IDEGF=IDEGF+1
      WRITE(6,6013)
 6013 FORMAT('OSCALE OF THE COORDINATE SYSTEM DEFINED BY INNER ADJUSTMENT
     1T PROCEEDURE.')
      GO TO 10
C 14  COMPLETELY FIX ONE OR MORE COORDINATES OF A STATION
 1400 CONTINUE
      ASSIGN 1410 TO JRTN
      GO TO 960
 1410 CONTINUE
      READ(3) IS,ISTA,XI,W
      DO 1420 I=1,3
 1420 DIS(I)=XI(I)-STAUVW(I,ISTA)
      IB=LOC(NBLOCK-1)+ISTA
      NB=LOC(NBLOCK)
      DO 1430 I=1,3
C   THE DIAGONAL ELEMENTS OF W ARE USED AS INDICATORS TO SHOW
C          WHICH COORDINATES ARE TO BE FIXED.
      IF(W(I,I).EQ.0.0) GO TO 1424
      REDN(I,I,IB)=1.0
      U(I,NBLOCK)=DIS(I)
      IDEGF=IDEGF+1
      L(IB)=1
      GO TO 1430
 1424 REDN(I,I,NB)=1.0
      L(NB)=1
```

```
 1430 CONTINUE
      WRITE(6,6014) IS,(STANAM(I,ISTA),I=1,5),XI,(W(I,I),I=1,3)
 6014 FORMAT(////15X,'CARTESIAN COORDINATES OF STATION',I5,2X,5A4,3X,
     1'FIXED AT'//15X,3F16.2//15X,'FIXED COORDINATES ARE INDICATED BY
     2NON ZERO ENTRY BELOW'//15X,3F16.2)
      L(LOC(ISTA))=1
      GO TO 10
C
 1500 CONTINUE
C 15 COMPLETELY FIX ONE OR MORE COORDINATES OF RELATIVE POSITION
      ASSIGN 1510 TO JRTN
      GO TO 960
 1510 CONTINUE
      READ (3) IS,ISTA,JS,JSTA,DXB,W
      IB=LOC(NBLOCK-1)+ISTA
      JB=LOC(NBLOCK-1)+JSTA
      NB=LOC(NBLOCK)
      L(IB)=1
      L(JB)=1
      L(LOC(ISTA))=1
      L(LOC(JSTA))=1
      DO 1530 I=1,3
      IF(W(I,I).EQ.0.0) GO TO 1524
      REDN(I,I,IB)=1.0
      REDN(I,I,JB)=-1.0
      U(I,NBLOCK)=DXB(I)-(STAUVW(I,ISTA)-STAUVW(I,JSTA))
      IDEGF=IDEGF+1
      GO TO 1530
 1524 REDN(I,I,NB)=1.0
      L(NB)=1
 1530 CONTINUE
      WRITE(6,6015) IS,(STANAM(I,ISTA),I=1,5),JS,(STANAM(I,JSTA),I=1,5),
     1 DXB,(W(I,I),I=1,3)
 6015 FORMAT(////15X,'RELATIVE POSITION BETWEEN STATION',I5,3X,5A4/
     137X,'AND STATION',I5,3X,5A4/          15X,
     2'FIXED AT'/15X,3F16.2//15X,'RELATIVE COORDINATES WHICH ARE FIXED
     3ARE INDICATED BY A NON-ZERO ENTRY BELOW'//15X,3F16.2)
      GO TO 10
 1600 CONTINUE
C  SET UP A BLOCK OF 3 DUMMY EQUATIONS
      ASSIGN 1610 TO JRTN
      GO TO 960
 1610 CONTINUE
      NB=LOC(NBLOCK)
      DO 1615 I=1,3
 1615 REDN(I,I,NB)=1.0
      L(NB)=1
      WRITE(6,6016)
 6016 FORMAT('0BLOCK OF 3 DUMMY EQUATIONS ADDED TO REDUCED NORMALS.')
      GO TO 10
C
C   EXPAND REDUCED NORMALS BY ADDING A BLOCK OF THREE LAGRANGE MULTIPIERS
  960 CONTINUE
      NBLOCK=NBLOCK+1
      IF(NBLOCK.LE.MAXBLK) GO TO 961
      WRITE(6,6096)
 6096 FORMAT('0ATTEMPTED CONSTRAINT RESULTS IN AN ATTEMPT TO EXPAND THE
     1REDUCED NORMAL EQUATION MATRIX'/15X,'BEYOND ITS DIMENSIONS.'/
```

```
     215X,'PROGRAM STOPS')
      STOP
  961 CONTINUE
      DO 965 ISTA=1,NBLOCK
      NB=LOC(NBLOCK-1)+ISTA
      L(NB)=0
      DO 965 I=1,3
      DO 965 J=1,3
  965 REDN(I,J,NB)=0.0
      DO 966 I=1,3
  966 U(I,NBLOCK)=0.0
      GO TO JRTN,(1110,1210,1310,1410,1510,1610)
C
C
 1700 CONTINUE
 1800 CONTINUE
 1900 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONAP
      IMPLICIT REAL*8(A-H,O-Z)
C   PROCESS A PRIORI CONSTRAINTS ON, AND BETWEEN, STATIONS
      COMMON/NSTA/NSTA,NBLOCK
      INTEGER*2 L,LSOLVE,IDS
      INTEGER ENDSIG/1HE/,CONTIN,STANAM
      COMMON/NORMEQ/REDN(3,3,2485),U(3,70),L(2485),LSOLVE
      COMMON/WPW/WPW,XPU,IDEGF,IFSTA
      LOC(K)=(K*(K+1))/2
      IFSTA=0
      NBLOCK=NSTA
   10 CONTINUE
C     WRITE(6,6801) NSTA,NBLOCK
 6801 FORMAT(7I7)
      READ (3) KODE,CONTIN
      IF(CONTIN.EQ.ENDSIG) GO TO 1000
      IF(KODE.LE.0) GO TO 950
      IF(KODE.GT.19) GO TO 950
      IF(KODE.GT.9) GO TO 11
      GO TO (100,200,300,400,500,600,700,800,900),KODE
   11 IF(KODE.LT.11) GO TO 950
      KODE2=KODE-10
      GO TO (1100,1200,1300,1400,1500,1600,1700,1800,1900),KODE2
C
C
  100 CONTINUE
  200 CONTINUE
  300 CONTINUE
  400 CONTINUE
  500 CONTINUE
      CALL CONAP1(KODE)
      GO TO 10
  600 CONTINUE
  700 CONTINUE
  800 CONTINUE
  900 CONTINUE
      GO TO 950
C
 1100 CONTINUE
 1200 CONTINUE
 1300 CONTINUE
 1400 CONTINUE
 1500 CONTINUE
 1600 CONTINUE
      CALL CONAP2(KODE2)
      GO TO 10
 1700 CONTINUE
 1800 CONTINUE
 1900 CONTINUE
      GO TO 950
C
  950 WRITE(6,6095) KODE
 6095 FORMAT('0ILLEGAL CONSTRAINT CODE IN CONAP IGNORED',I5)
      GO TO 10
C
CHECK TO SEE IF NORMALS ARE SOLVABLE
 1000 CONTINUE
      LSOLVE=1
```

```
      DO 1010 ISTA=1,NSTA
      NB=LOC(ISTA)
C     WRITE(6,6801) NB,L(NB)
      IF(L(NB).NE.0) GO TO 1010
      LSOLVE=0
      IDEGF=ISTA
 1010 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SOLVE
C   SOLVE NORMAL EQUATIONS AND COMPUTE INVERSE FOR UP TO 40 STATIONS
C     BY THE METHOD OF TRIANGULAR MATRICES.
C   SEE DEPT. OF GEODETIC SCIENCE REPORT NO. 86, SECTION 5
C   THE SCHEME USED TO ADDRESS THE UPPER TRIANGULAR PART OF THE REDUCED
C     NORMALS IS THE SAME AS THAT USED IN FORMRN.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 L,LSOLVE
      COMMON/NORMEQ/REDN(3,3,2485),U(3,70),L(2485),LSOLVE
      COMMON/WPW/WPW,XPU,IDEGF,NFSTA
      COMMON/NSTA/NSTA1,NBLOCK
      DIMENSION DX(3,70),TEMP(3,3,70),TA(3,3),TB(3,3),TC(3)
      INTEGER*2PCODE(20)
      COMMON/PCODES/PCODE
      INTEGER*2 PIVOT(210)
      DIMENSION RN(22365),RU(210),TEM(210),DXI(210)
      EQUIVALENCE (RN,REDN),(RU,U,DXI),(TEM,TEMP)
      EQUIVALENCE(DX,U)
C
      LOC(K)=(K*(K+1))/2
C
      IF(PCODE(16).LT.2) GO TO 5
      DO 2 ISTA=1,NBLOCK
      DO 1 JSTA=ISTA,NBLOCK
      NB=LOC(JSTA-1)+ISTA
      WRITE(6,6803) ISTA,JSTA,NB,L(NB)
 6803 FORMAT(1H0,7I7)
      WRITE(6,6801) ((REDN(I,J,NB),J=1,3),I=1,3)
 6801 FORMAT(//3(3D20.8/))
    1 CONTINUE
      WRITE(6,6802) (U(I,ISTA),I=1,3)
 6802 FORMAT(/3D20.8)
    2 CONTINUE
    5 CONTINUE
CHECK TO SEE IF THIS SET OF EQUATIONS HAS BEEN MARKED SOLVABLE
      IF(LSOLVE.GE.1) GO TO 10
    9 WRITE(6,6001)    IDEGF
 6001 FORMAT('OREDUCED NORMALS MARKED UNSOLVABLE. PROGRAM STOPS.',I5)
      STOP
   10 CONTINUE
      REWIND 2
C    NSTA1 GIVES THE NUMBER OF GROUND STATIONS IN THE ADJUSTMENT
C    NBLOCK GIVES THE TOTAL NUMBER OF BLOCKS OF UNKNOWNS IN THE REDUCED NORMALS,
C        INCLUDING BOTH STATION COORDINATES AND BLOCKS OF LAGRANGE NULTIPLIERS.
C
C    THE EXPANDED (NBLOCK SQUARE) SET OF REDUCED NORMALS IS SOLVED
      NSTA=NBLOCK
      REWIND 2
      DO 20 ISTA=1,NBLOCK
      DO 20 JSTA=ISTA,NBLOCK
      NB=LOC(JSTA-1)+ISTA
   19 WRITE(2) ((REDN(I,J,NB),I=1,3),J=1,3)
   20 CONTINUE
      REWIND 2
      NUNK=3*NBLOCK
      NB=LOC(NUNK)
      DO 25 I=1,NB
```

```
   25 RN(I)=0.0
      DO 30 ISTA=1,NBLOCK
      DO 30 JSTA=ISTA,NBLOCK
      READ(2) TA
      DO 30 I=1,3
      II=3*(ISTA-1)+I
      DO 30 J=1,3
      JJ=3*(JSTA-1)+J
      IF(II.GT.JJ) GO TO 30
      IB=LOC(JJ-1)+II
      RN(IB)=TA(I,J)
   30 CONTINUE
C
      IF(PCODE(16).LT.3) GO TO 41
      DO 40 I=1,NUNK
      DO 35 J=I,NUNK
      NB=LOC(J-1)+I
      TEM(J)=RN(NB)
   35 CONTINUE
   40 WRITE(6,6805) I,(TEM(J),J=I,NUNK)
 6805 FORMAT(I5,6D19.10/250(5X,6D19.10/))
   41 CONTINUE
C
C
C PERFORM FIRST REDUCTION - COMPUTE R AND C MATRICES   -EQ.5-9
C
      DO 100 I=1,NUNK
C
C   FIND PIVOT ELEMENT
      IP=0
      PMAX=0.0
      DO 55 J=I,NUNK
       NB=LOC(J)
      IF(DABS(RN(NB)).LE.PMAX) GO TO 55
      PMAX=DABS(RN(NB))
      IP=J
   55 CONTINUE
      PIVOT(I)=IP
      IF(PCODE(16).GT.0) WRITE(6,6806) I,IP,PMAX
 6806 FORMAT(2I7,D20.10)
      IF(IP.EQ.0) GO TO 9
C   SWITCH ROWS AND COLUMNS
      CALL SWITCH(I,IP)
C
      IB=LOC(I)
      RN(IB)=1.0/RN(IB)
      IF(I.EQ.NUNK) GO TO 100
      IP1=I+1
C   OUTER LOOP  --REDUCE ROW K
      DO 80 K=IP1,NUNK
      KIB=LOC(K-1)+I
C   GET MULTIPLIER
      TD=RN(KIB)*RN(IB)
      IF(TD.EQ.0.0)GO TO 80
C    REDUCE CONSTANT COLUMN
      RU(K)=RU(K)-TD*RU(I)
C   INNER LOOP
      DO 80 J=K,NUNK
```

```
      NB=LOC(J-1)+I
      KJB=LOC(J-1)+K
      RN(KJB)=RN(KJB)-TD*RN(NB)
   80 CONTINUE
  100 CONTINUE
C
C
      XPU=0.0
      DO 300 NMIP1=1,NUNK
      I=NUNK-NMIP1+1
      NB=LOC(I)
C  ACCUMULATE XPU
      XPU=XPU+RN(NB)*RU(I)**2
      IP1=I+1
C
C  BACK SUBSTITUATION
      IF(I.EQ.NUNK) GO TO 190
      DO 150 J=IP1,NUNK
      IB=LOC(J-1)+I
      RU(I)=RU(I)-RN(IB)*DXI(J)
  150 CONTINUE
  190 CONTINUE
      DXI(I)=RN(NB)*RU(I)
C
C
C   DEVELOP ROW I OF INVERSE MATRIX
      IF(I.EQ.NUNK) GO TO 300
      DO 280 NMJP1=1,NMIP1
      J=NUNK-NMJP1+1
      TD=0.0
      IF(I.EQ.J) TD=1.0
      DO 270 K=IP1,NUNK
      IB=LOC(K-1)+I
      IF(J.EQ.I) GO TO 245
      IF(K.GT.J) GO TO 240
      JB=LOC(J-1)+K
      TE=RN(JB)
      GO TO 250
  240 CONTINUE
      JB=LOC(K-1)+J
      TE=RN(JB)
      GO TO 250
  245 CONTINUE
      TE=TEM(K)
  250 CONTINUE
      TD=TD-RN(IB)*TE
  270 CONTINUE
C   STORE ITH ROW TEMPORARILY IN TEM
      TEM(J)=RN(NB)*TD
  280 CONTINUE
C
COPY ITH ROW OUT OF TEM
      DO 290 NMJP1=1,NMIP1
      J=NUNK-NMJP1+1
      JB=LOC(J-1)+I
      RN(JB)=TEM(J)
  290 CONTINUE
C
```

```
C    UNDO PIVOTING
      IP=PIVOT(I)
      CALL SWITCH(I,IP)
  300 CONTINUE
      REWIND 2
C
      DO 305 I=1,NUNK
      DO 304 J=I,NUNK
      IB=LOC(J-1)+I
  304 TEM(J)=RN(IB)
      WRITE(2) (TEM(J),J=I,NUNK)
      IF(PCODE(16).LT.3) GO TO 305
      WRITE(6,6805) I,(TEM(J),J=I,NUNK),DXI(I)
  305 CONTINUE
C
C
      REWIND 2
      DO 310 I=1,NUNK
      READ(2) (TEM(J),J=I,NUNK)
      DO 310 J=I,NUNK
      ISTA=(I-1)/3+1
      JSTA=(J-1)/3+1
      NB=LOC(JSTA-1)+ISTA
      II=I-3*(ISTA-1)
      JJ=J-3*(JSTA-1)
      REDN(II,JJ,NB)=TEM(J)
      IF(ISTA.EQ.JSTA) REDN(JJ,II,NB)=TEM(J)
  310 CONTINUE
      REWIND 2
C
C   OUTPUT THE SOLUTION AND COVARIANCE BLOCKS CORRESPONDING TO THE STATION UNKNOWN
      DO 320 ISTA=1,NSTA1
      NB=LOC(ISTA)
      DO 320 I=1,3
      TEMP(I,I,ISTA)=REDN(I,I,NB)
  320 CONTINUE
      WRITE(2) ((TEMP(I,I,ISTA),I=1,3),ISTA=1,NSTA1)
      DO 350 ISTA=1,NSTA
      IF(ISTA.GT.NSTA1) GO TO 339
      DO 330 JSTA=ISTA,NSTA1
      NB=LOC(JSTA-1)+ISTA
      DO 330 I=1,3
      DO 330 J=1,3
      TEMP(I,J,JSTA)=REDN(I,J,NB)
  330 CONTINUE
      WRITE(2) (DX(I,ISTA),I=1,3),(((TEMP(I,J,JSTA),I=1,3),J=1,3),
     1JSTA=ISTA,NSTA1)
  339 CONTINUE
      IF(PCODE(16).LT.2) GO TO 350
      IB=LOC(ISTA)
      WRITE(6,6002) ISTA,(DX(I,ISTA),I=1,3),((REDN(I,J,IB),
     1 J=1,3),I=1,3)
 6002 FORMAT(///I5/3F16.8//3(3D16.8/))
      DO 340 JSTA=ISTA,NSTA
      NB=LOC(JSTA-1)+ISTA
  340 WRITE(6,6003) JSTA,((REDN(I,J,NB                    ),J=1,3),I=1,3)
 6003 FORMAT(//I5//3(3D16.8/))
  350 CONTINUE
```

```
REWIND 2
RETURN
END
```

```
      SUBROUTINE SWITCH(I,IP)
C     SWITCH ROW AND COLUMN I WITH ROW AND COLUMN IP
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*2 L,LSOLVE
      COMMON/NORMEQ/REDN(3,3,2485),U(3,70),L(2485),LSOLVE
      COMMON/NSTA/NSTA1,NBLOCK
      DIMENSION RN(22365),RU(210)
      EQUIVALENCE (RN,REDN),(RU,U)
      LOC(K)=(K*(K+1))/2
      NSTA=NBLOCK
      NUNK=3*NBLOCK
      IF(IP.EQ.I) RETURN
      DO 25 J=1,NUNK
      NB=LOC(J-1)+I
      IF(J.EQ.I) GO TO 22
      IF(J-IP) 16,25,18
   16 IB=LOC(IP-1)+J
      GO TO 24
   18 IB=LOC(J-1)+IP
      GO TO 24
   22 IB=LOC(IP)
      GO TO 24
   24 CONTINUE
      TD=RN(IB)
      RN(IB)=RN(NB)
      RN(NB)=TD
   25 CONTINUE
      TD=RU(IP)
      RU(IP)=RU(I)
      RU(I)=TD
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION ANRADD(ISGN,IDEG,MIN,SEC)
      INTEGER*2 MINUS/1H-/, PLUS/1H+/,AMPSAN/1H&/,ISGN,IDEG,MIN
      DOUBLE PRECISION SEC
      IF(IDEG.GE.0) GO TO 10
      ISGN=MINUS
      IDEG=-IDEG
   10 CONTINUE
      ANRADD=(DFLOAT((IDEG*60+MIN)*60)+SEC)/206264.80625
      IF(ISGN.EQ.MINUS)ANRADD=-ANRADD
      IF(ISGN.EQ.AMPSAN) ISGN=PLUS
      RETURN
      END
```

```
      INTEGER FUNCTION KSTAID(ID)
      COMMON/STAORD/KORDER(150)
      COMMON/NSTA/NSTA
      KSTAID=0
C   SEARCH TABLE OF STATION IDENTIFIERS FOR THE INTERNAL NUMBER OF THIS STATION
      DO 10 I=1,NSTA
      IF(KORDER(I).NE.ID) GO TO 10
      KSTAID=I
      RETURN
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINEUVWTG3(UVW,DATUM,PHI,LAM,H)
C     CONVERT RECTANGULAR TO GEODETIC COORDINATES
C     ALIAS FOR UVWTG
      IMPLICIT REAL*8(A-Z)
      DIMENSION UVW(3),DATUM(2)
      LAM=DATAN2(UVW(2),UVW(1))
      IF(LAM.LT.0.0) LAM=LAM+6.28318530717958
      OME2=(DATUM(2)/DATUM(1))**2
      E2=1.0-OME2
      P=DSQRT(UVW(1)**2+UVW(2)**2)
      WP=UVW(3)/P
      TP1=WP/OME2
      PHI1=DATAN(TP1)
    5 TTP=TP1*TP1
      SECP=DSQRT(1.0+TTP)
      N=DATUM(1)*SECP/DSQRT(1.0+OME2*TTP)
      H=P*SECP-N
      TP2=WP/(1.0-E2*N/(N+H))
      PHI=DATAN(TP2)
      IF(DABS(PHI-PHI1).LT.1.D-12) RETURN
      PHI1=PHI
      TP1=TP2
      GO TO 5
      END
```

```
      SUBROUTINE UVWTG2(UVW,DATUM,PHI,LAM,H)
C     CONVERT RECTANGULAR TO GEODETIC COORDINATES
C     ALIAS FOR UVWTG
      IMPLICIT REAL*8(A-Z)
      DIMENSION UVW(3),DATUM(2)
      LAM=DATAN2(UVW(2),UVW(1))
      IF(LAM.LT.0.0) LAM=LAM+6.28318530717958
      OME2=(DATUM(2)/DATUM(1))**2
      E2=1.0-OME2
      P=DSQRT(UVW(1)**2+UVW(2)**2)
      WP=UVW(3)/P
      TP1=WP/OME2
      PHI1=DATAN(TP1)
    5 TTP=TP1*TP1
      SECP=DSQRT(1.0+TTP)
      N=DATUM(1)*SECP/DSQRT(1.0+OME2*TTP)
      H=P*SECP-N
      TP2=WP/(1.0-E2*N/(N+H))
      PHI=DATAN(TP2)
      IF(DABS(PHI-PHI1).LT.1.D-12) RETURN
      PHI1=PHI
      TP1=TP2
      GO TO 5
      END
```

```
      SUBROUTINE DELL(DX,DXCOV, PHI,LAM,H,DATUM,DP,DL,DH,DELCOV,RLX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DX(3),DXCOV(3,3),DATUM(2),DELCOV(3,3),DE(3),GE(3,3)
      DIMENSION RLX(3,3)
      REAL*8 LAM
      ESQ=1.0-(DATUM(2)/DATUM(1))**2
      CP=DCOS(PHI)
      SP=DSIN(PHI)
      SL=DSIN(LAM)
      CL=DCOS(LAM)
      EW=DSQRT(1.0-ESQ*SP**2)
      EN=DATUM(1)/EW
      EM=EN*(1.0-ESQ)/EW**2
      H1=EM+H
      H2=(EN+H)*CP
      GE(1,1)=-SP*CL/H1
      GE(1,2)=-SP*SL/H1
      GE(1,3)=CP/H1
      GE(2,1)=-SL/H2
      GE(2,2)=CL/H2
      GE(2,3)=0.0
      GE(3,1)=CP*CL
      GE(3,2)=CP*SL
      GE(3,3)=SP
C
      CALL DGMPRD(GE,DX,DE,3,3,1)
      DP=DE(1)
      DL=DE(2)
      DH=DE(3)
C
      DO 14 I=1,3
      DO 14 J=1,3
      SUM=0.0
      DO 12 K=1,3
      DO 12 L=1,3
   12 SUM=SUM+GE(I,K)*DXCOV(K,L)*GE(J,L)
   14 DELCOV(I,J)=SUM
C
      DO 15 J=1,3
      RLX(1,J)=GE(1,J)*H1
      RLX(2,J)=GE(2,J)*H2
   15 RLX(3,J)=GE(3,J)
C
      RETURN
      END
```

```
SUBROUTINE DANG(ANGR,ISGN,IDEG,MIN,SEC)
IMPLICIT REAL*8(A-H,O,Z)
INTEGER BLANK/1H /,MINUS/1H-/
ISGN=BLANK
IF(ANGR.LT.0.0) ISGN=MINUS
ANGD=57.295779513082D0*DABS(ANGR)
IDEG=IDINT(ANGD)
FMIN=ANGD-DFLOAT(IDEG)
FMIN=FMIN*60.0
MIN=IDINT(FMIN)
SEC=(FMIN-DFLOAT(MIN))*60.0
RETURN
END
```

```
      SUBROUTINE UVWTG(UVW,DATUM,PHI,LAM,H)
C     CONVERT RECTANGULAR TO GEODETIC COORDINATES
      IMPLICIT REAL*8(A-Z)
      DIMENSION UVW(3),DATUM(2)
      LAM=DATAN2(UVW(2),UVW(1))
      IF(LAM.LT.0.0) LAM=LAM+6.28318530717958
      OME2=(DATUM(2)/DATUM(1))**2
      E2=1.0-OME2
      P=DSQRT(UVW(1)**2+UVW(2)**2)
      WP=UVW(3)/P
      TP1=WP/OME2
      PHI1=DATAN(TP1)
    5 TTP=TP1*TP1
      SECP=DSQRT(1.0+TTP)
      N=DATUM(1)*SECP/DSQRT(1.0+OME2*TTP)
      H=P*SECP-N
      TP2=WP/(1.0-E2*N/(N+H))
      PHI=DATAN(TP2)
      IF(DABS(PHI-PHI1).LT.1.D-12) RETURN
      PHI1=PHI
      TP1=TP2
      GO TO 5
      END
```

```
      FUNCTION KSID2(IS)
      KSID2=KSTAID(IS)
      IF(KSID2.GT.0) RETURN
      WRITE(6,6000) IS
 6000 FORMAT(//10X,'STATION NUMBER NOT FOUND IN INPUT LIST',I5)
      STOP
      END
```

```
      SUBROUTINE SATXYZ (XS,YS,ZS)
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER STANAM,IDS*2
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/STAPLH/STAPLH(2,150)
      COMMON/RANGED/RAN(50),VARRA(50),RMSMC,NSTE,KSTATE(50),ITEST,NITR
      DIMENSION A( 3),AN(3,3),C(3),L(3),M(3),AX(3)
C   GET THE FIRST APPROXIMATION TO THE SATELLITE POSITION
      PI=3.14159265358
      TPI=2.*PI
      PHI=0.0
      ALAM=0.0
      IS=NSTE
      DO 20 I=1,IS
      PHI=PHI+STAPLH(1,KSTATE(I))
      STALNG=STAPLH(2,KSTATE(I))
      IF(I.EQ.1) GO TO 19
      IF(STALNG-STAPLH(2,KSTATE(1)).GT.PI)STALNG=STALNG-TPI
      IF(STAPLH(2,KSTATE(1))-STALNG.GT.PI)STALNG=STALNG+TPI
   19 ALAM=ALAM + STALNG
   20 CONTINUE
      PHI=PHI/IS
      ALAM=ALAM/IS
      H=1.6D06
      IDTS=IDS(KSTATE(1))
      CALL UVWD(DATPRM(1,IDTS),DATPRM(2,IDTS),PHI,ALAM,H,XS,YS,ZS)
C
      NITR=0
   25 CONTINUE
C    START ANOTHER ITERATION
      NITR=NITR+1
      WPW=0.0
      DO 30 I=1,3
      C(I)=0.0
      DO 30 J=1,3
   30 AN(I,J)=0.0
C
      DO 50 IS=1,NSTE
      DX=XS-STAUVW(1,KSTATE(IS))
      DY=YS-STAUVW(2,KSTATE(IS))
      DZ=ZS-STAUVW(3,KSTATE(IS))
      R=DSQRT(DX*DX+DY*DY+DZ*DZ)
      AL=RAN(IS)-R
      WPW=AL*AL/VARRA(IS)**2+WPW
      A(1)=DX/R
      A(2)=DY/R
      A(3)=DZ/R
      DO 40 I=1,3
      C(I)=C(I)+A(I)*AL/VARRA(IS)**2
      DO 40 J=1,3
   40 AN(I,J)=AN(I,J)+A(I)*A(J)/VARRA(IS)**2
   50 CONTINUE
      CALL DMINV(AN,3,DET,L,M)
      CALL DGMPRD(AN,C,AX,3,3,1)
      RMSMC=WPW/(NSTE-3)
C   TEST FOR CONVERGENCE
      ICONVR=1
```

```
      DO 55 I=1,3
      IF(DABS(AX(I)).GT.0.01) ICONVR=0
   55 CONTINUE
C
C  UPDATE
      XS=XS+AX(1)
      YS=YS+AX(2)
      ZS=ZS+AX(3)
      IF(ICONVR.EQ.1) RETURN
      IF(NITR.LT.20) GO TO 25
C   SET ITEST =2 INDICATED THAT CONVERGENCE WAS NOT OBTAINED IN 20 ITERATIONS
      ITEST=2
      RETURN
      END
```

```
      SUBROUTINE RNG360
      IMPLICIT REAL*8 (A-H,O-Z)
C   S/360 VERSION OF SAR PROGRAM FOR SATELLITE DISTANCES
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      INTEGER*4 ENDSIG/1HE/,CONTIN,DELCOD(2)/1H ,1H*/,ECODE
      INTEGER*2 ID(50),IYR(50),IDAY(50),IHR(50),MIN(50)
      DIMENSION SEC(50),RAN(50),VARRA(50),MONTH(50),KSTATE(50)
      DIMENSION DN(3,3,150),BN(3,3,50),DDN(3,3),DK(3,150),DDK(3),A(3)
      COMMON/NSTA/NSTA
      DIMENSION NOBSTA(150),VPVSTA(150)
      COMMON/RANGED/RAN,VARRA,RMSMC,NSTE,KSTATE,ITEST,NITR
      DIMENSION POSSAT(3),DX(3)
      DIMENSION L1(3),L2(3)
      INTEGER STANAM,IDS*2
      COMMON/STALOC/STAUVW(3,150),DATPRM(2,15),DATNAM(4,15),
     1STANAM(5,150),IDS(150)
      COMMON/STAORD/KORDER(150)
      COMMON/WPW/WPW,XPU,IDEGF,NFSTA
      COMMON/STAPLH/STAPLH(2,150)
      DATA RPD/57.295779513/
C
      REWIND 2
      REWIND 3
      READ (3) TD,STAPLH
      WRITE(6,6004) TD
 6004 FORMAT(//20X,'TEST VARIANCE =',F20.2)

      WRITE(6,6001)
 6001 FORMAT(///'0STATION',T12,'DATE',T24,'TIME',T43,'RANGE',T60,
     1  'UNCERTAINTY',T76,'MISCLOSURE')
C
C
      DO 70 KSTA=1,NSTA
      NOBSTA(KSTA)=0
      VPVSTA(KSTA)=0.0
      DO 70 I=1,3
      DK(I,KSTA)=0.0
      DO 70 J=1,3
      DN(I,J,KSTA)=0.0
   70 CONTINUE
C
      KEVENT=0
      EPR=0.0
  210 CONTINUE
      READ (3) IEVENT,NSTE,EPR, (ID(IS),IYR(IS),MONTH(IS),IDAY(IS),
     X  IHR(IS),
     1MIN(IS),SEC(IS),RAN(IS),VARRA(IS),KSTATE(IS),IS=1,NSTE),CONTIN
      WRITE(6,6008) IEVENT
 6008 FORMAT(/ 1X,'EVENT',I6)
      ITEST=0
      IF(NSTE.GT.3) GO TO 220
C   SET ITEST =1 TO INDICATE THAT LESS THAN FOUR STATIONS WERE OBSERVING
      ITEST=1
      DO 280 IS=1,NSTE
  280 WRITE(6,6009) ID(IS),IYR(IS),MONTH(IS),IDAY(IS),IHR(IS),
     1MIN(IS),SEC(IS),RAN(IS),VARRA(IS)
```

```
 6009 FORMAT(I7,2X,I2,1X,A3,1X,I2,2X,2I3,F8.4,F18.3,F15.2,F15.2)
      GO TO 630
  220 CONTINUE
      CALL SATXYZ(XS,YS,ZS)
      IF(RMSMC.GT.TD) ITEST=3
      IF(ITEST) 311,300,311
C
C
C     SET UP OBSERVATION EQUATIONS FOR THIS EVENT AND COMPUTE
C     CONTRIBUTIONS TO THE NORMAL EQUATIONS
  300 CONTINUE
      KEVENT=KEVENT+1
      DO 310 I=1,3
      DDK(I)=0.0
      DO 310 J=1,3
      DDN(I,J)=0.0
  310 CONTINUE
C
  311 CONTINUE
      POSSAT(1)=XS
      POSSAT(2)=YS
      POSSAT(3)=ZS
      DO 390 IS=1,NSTE
      KSTA=KSTATE(IS)
      DO 305 I=1,3
  305 DX(I)=POSSAT(I)-STAUVW(I,KSTA)
      RC=DSQRT(DPDOT(DX,DX,3))
      AL=RAN(IS)-RC
      WRITE(6,6009) ID(IS),IYR(IS),MONTH(IS),IDAY(IS),IHR(IS),
     1MIN(IS),SEC(IS),RAN(IS),VARRA(IS),AL
      IF(ITEST) 390,307,390
  307 CONTINUE
COMPUTE WEIGHT
      WT=1./VARRA(IS)**2
      DO 306 I=1,3
  306 A(I)=DX(I)/RC
C
COMPUTE VPV OF MISCLOSURES
      VPVTO=AL*WT*AL
      VPVSTA(KSTA)=VPVSTA(KSTA)+VPVTO
      NOBSTA(KSTA)=NOBSTA(KSTA)+1
COMPUTE CONTRIBUTIONS TO NORMAL EQUATIONS
      DO 330 I=1,3
      TERM=A(I)*WT*AL
      DK(I,KSTA)=DK(I,KSTA)-TERM
      DDK(I)=DDK(I)+TERM
      DO 330 J=1,3
      TERM=A(I)*WT*A(J)
      BN(I,J,IS)=-TERM
      DN(I,J,KSTA)=DN(I,J,KSTA)+TERM
      DDN(I,J)=DDN(I,J)+TERM
  330 CONTINUE
  390 CONTINUE
C
      IF(ITEST) 600,391,600
  391 CONTINUE
      CALL DMINV(DDN,3,DET,L1,L2)
      WRITE(2) NSTE,  DDN,DDK,(((BN(I,J,IS),I=1,3),J=1,3),KSTATE(IS),
```

```
     1IS=1,NSTE),CONTIN
  600 CONTINUE
      WRITE(6,6011)RMSMC,NITR
 6011 FORMAT('     VARIANCE OF EVENT ADJUSTMENT =',F10.2,
     1'  AFTER',I3,' ITERATIONS')
      IF(PCODE(11)) 610,630,610
  610 IF(PCODE(11)-3) 611,612,611
  611 WRITE(6,6022) POSSAT
 6022 FORMAT('  SATELLITE POSITION',3F15.3)
      IF(PCODE(11)-2) 612,630,612
  612 IDTS=IDS(KSTATE(1))
      CALL UVWTG2(POSSAT,DATPRM(1,IDTS),PHI,FLAM,H)
      PHI=PHI*RPD
      FLAM=FLAM*RPD
      WRITE(6,6023) PHI,FLAM,H
 6023 FORMAT('  GEOD. COORD. OF SATELLITE',2F14.6,F14.1)
  630 CONTINUE
      IF(ITEST) 290,640,290
  290 WRITE(6,6015) ITEST
 6015 FORMAT(1H ,27X,'ENTIRE EVENT DELETED.   DELETION CODE = ',I3)

C    ITEST=0 MEANS A GOOD EVENT
C    ITEST=1 MEANS NOT ENOUGH OBSERVATIONS
C    ITEST=2 MEANS MORE THAN 20 ITERATIONS WERE REQUIRED TO GET APPROXIMATE
C    SATELLITE POSITION
C     ITEST=3 MEANS THE EVENT IS REJECTED BECAUSE THE EVENT VARIANCE IS GREATER
C        THAN THE TEST VARIANCE
  640 CONTINUE
C
      IF(CONTIN.EQ.ENDSIG) GO TO 700
      GO TO 210
C
C
  700 CONTINUE
C
CHECK TO SEE IF END SIGNAL HAS BEEN WRITEEN ON DATA SET FT02
      IF(ITEST.EQ.0) GO TO 710
      BACKSPACE 2
C    READ AND WRITE LAST RECORD FROM LSST GOOD EVENT
      READ (2) NSTE,  DDN,DDK,(((BN(I,J,IS),I=1,3),J=1,3),KSTATE(IS),
     1IS=1,NSTE)
      WRITE(2) NSTE,  DDN,DDK,(((BN(I,J,IS),I=1,3),J=1,3),KSTATE(IS),
     1IS=1,NSTE),CONTIN
  710 CONTINUE
      WRITE(2)       ((( DN(I,J,KSTA),I=1,3),DK(J,KSTA),J=1,3),
     XKSTA=1,NSTA)
C     WRITE(6,6018)(KORDER(KSTA),((DN(I,J,KSTA),J=1,3),I=1,3),
C    1KSTA=1,NSTA)
 6018 FORMAT((I5/3(3D18.7/)))
      WPW=0.0
      NOBS=0
      WRITE(6,6019)
 6019 FORMAT(1H1,8(/),10X,'ANALYSIS OF MISCLOSURES BY STATION'//
     1T10,'STATION',T20,'NUMBER OF OBSERVATIONS',T50,'RMS MISCLOSURE')
      DO 750 KSTA=1,NSTA
```

```
      NOBS=NOBS+NOBSTA(KSTA)
      WPW=WPW+VPVSTA(KSTA)
      RMSMC=0.0
      IF(NOBSTA(KSTA).GT.0) RMSMC=DSQRT(VPVSTA(KSTA)/DFLOAT(NOBSTA(KSTA)
     1))
      WRITE(6,6020) KORDER(KSTA),NOBSTA(KSTA),RMSMC
 6020 FORMAT(T10,I7,T35,I7,T50,F14.2)
  750 CONTINUE
      IDEGF=NOBS-3*KEVENT
      RMSMC=DSQRT(WPW/DFLOAT(IDEGF))
      WRITE(6,6021) NOBS,KEVENT,IDEGF,WPW,RMSMC
 6021 FORMAT(////10X,'TOTAL NUMBER OF GOOD OBSERVATIONS',T60,I8//
     110X,'TOTAL NUMBER OF GOOD EVENTS',T60,I8,//

     210X,'CORRESPONDING DEGREES OF FREEDOM',T60,I8//
     310X,'TOTAL SUM OF SQUARES OF MISCLOSURES',T60,F11.2//
     410X,'CORRESPONDING STANDARD DEVIATION OF UNIT WEIGHT',T60,F11.2)
      RETURN
      END
```

```
      SUBROUTINE RRDATA
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 ENDSIG/1HE/,CONTIN
      INTEGER*2 ID(50),IYR(50),IDAY(50),IHR(50),MIN(50)
      DIMENSION SEC(50),RAN(50),VARRA(50),MONTH(50),KSTATE(50)
      COMMON/STAPLH/STAPLH(2,150)
      COMMON/OBSD/OBSD(150),OVOBSD
      INTEGER*2 PCODE(20)
      COMMON/PCODES/PCODE
      MAXSTE=50
      SPR=206264.80625
      PI=3.14159265358
      PI2=2.0*PI
      WPWSP=0.0
      READ(5,5004) TD,OVOBSD
      WRITE(6,6004) TD
 5004 FORMAT(F20.2,F10.2)
 6004 FORMAT(//20X,'TEST VARIANCE =',F20.2)


C
C    START DATA INPUT
      REWIND 3
      WRITE(3) TD,STAPLH
      IS=0
      IEVENT=0
      EPR=0.0
C    ENTER HERE FOR A NEW OBSERVATION
C
  200 IS=IS+1
  205 CONTINUE
      READ(5,1022,END=901)ID(IS),IYR(IS),MONTH(IS),IDAY(IS),IHR(IS),
     1MIN(IS),SEC(IS),SEC1,RAN(IS),RA1,VARRA(IS),CONTIN
      RAN(IS)=RAN(IS)+RA1/1000.
 1022 FORMAT(14X,I4,5I2,F2.0,F4.0,F16.0,F3.0,11X,F6.3,9X,A1)
      IF(SEC1.LT.1.) GO TO 201
      SEC(IS)=SEC(IS)+SEC1/10000.
      GO TO 202
  201 SEC(IS)=SEC(IS)+SEC1
  202 CONTINUE
      IF(CONTIN.EQ.ENDSIG) GO TO 250
      DDT=DFLOAT(MJD(IDAY(IS),MONTH(IS),IYR(IS)))
      DDT=DDT+(DFLOAT((IHR(IS)*60+MIN(IS))*60)+SEC(IS))/864.0D2
      IF(IS.LE.1) GO TO 210
C    THIS TEST SHOULD BE TRUE ONLY FOR THE FIRST CARD OF THE FIRST EVENT.
C
CHECK FOR END OF EVENT, ALLOWING 0.5 MS DISCREPANCY
      IF(DABS(DDT-EPR).GT.0.58D-8) GO TO 250
C
C    ENTER HERE TO BEGIN A NEW EVENT
C          THE FIRST ENTRY OF THE EVENT SHOULD ALWAYS BE MADE WITH IS=1
  210 CONTINUE
      IDD=ID(IS)
      KSTA=KSTAID(IDD)
      IF(KSTA.GT.0) GO TO 220
      WRITE(6,6042) ID(IS),IHR(IS),MIN(IS),SEC(IS),IDAY(IS),MONTH(IS),
     1IYR(IS)
 6042 FORMAT(5X, 'STATION NUMBER NOT FOUND IN INPUT LIST',I5,3X,2I3,
```

```
     1F8.4,3X,I3,A3,I2,'OBSERVATION IGNORED')
      GO TO 205
  220 CONTINUE
      IF(PCODE(12).EQ.1) VARRA(IS)=OBSD(KSTA)
      IF(PCODE(12).EQ.2) VARRA(IS)=OVOBSD
      KSTATE(IS)=KSTA
      EPR=DDT
      GO TO 200
C
C   END OF INPUT FOR THIS EVENT. BEGIN PROCESSING
  250 CONTINUE
      NSTE=IS-1
      IEVENT=IEVENT+1
      IF(IHR(NSTE+1).EQ.99) CONTIN=ENDSIG
      WRITE(3) IEVENT,NSTE,EPR, (ID(IS),IYR(IS),MONTH(IS),IDAY(IS),
     X  IHR(IS),
     1MIN(IS),SEC(IS),RAN(IS),VARRA(IS),KSTATE(IS),IS=1,NSTE),CONTIN
C   TEST FOR END OF INPUT
      IF(CONTIN.EQ.ENDSIG) GO TO 700
C   PREPARE FOR NEXT EVENT
      ID(1)=ID(NSTE+1)
      IYR(1)=IYR(NSTE+1)
      MONTH(1)=MONTH(NSTE+1)
      IDAY(1)=IDAY(NSTE+1)
      IHR(1)=IHR(NSTE+1)
      MIN(1)=MIN(NSTE+1)
      SEC(1)=SEC(NSTE+1)
      RAN(1)=RAN(NSTE+1)
      VARRA(1)=VARRA(NSTE+1)
C   RETURN TO START A NEW EVENT
      IS=1
      GO TO 210
C
  700 RETURN
C
C   ERROR EXITS
  901 CONTINUE
C   ENTER HERE IF END SIGNAL CARD IS MISSING FROM INPUT DATA DECK
      CONTIN=ENDSIG
      GO TO 250
      END
```